MeteoCAL Tools User's Manual: Draft 0.9

Changes from version 0.5
Appendix C has been updated to improve installation guidance.

Changes from version 0.6
Changed title to MeteoCAL Tools User's Guide
Added: Appendix E Localising the Interface
Appendix F Internationalisation/localisation Guidelines
New chapter (8) on linking to new modules/courses on a web server
Information on the UniRed editor to chapter 9 (previously chapter 8)
Improved: Appendix B MeteoCAL XML

Changes from version 0.7
Small changes in Chapter 8 to reflect new language structures
Added: Appendix E Localisation – Adding a new language
Modified: Appendix F Localisation – Setting which languages will be displayed
<matemtext> tag guidance added (allowing links to popups from questions)

Changes from version 0.8
Changes regarding the creation of new course files.

Introduction
This manual is intended for those whose task is either to maintain MeteoCAL modules and units of study or to create new modules and units of study. These tools were developed by Intrallect for EUMETSAT originally for MSG-CAL but as they are equally useful for many other meteorological courses the tools are collectively described as MeteoCAL tools. The term MSG-CAL is used to refer to the specific modules produced for training in aspects of the MSG satellite. The term MeteoCAL is used to describe the tools.

This manual includes:

Introduction 1
1: The Structure of MeteoCAL modules 3
2: Changing an existing Unit-of-Study 7
3: Creating a New MeteoCAL Module 13
4: Creating a New MeteoCAL Unit-of-Study 20
5: Modify the structure of an existing module 31
6: Server-side architecture 33
7: Transforming Source Files 36
8: Making Modules/Courses Visible on a Web Server 37
9: Choosing and Using an XML Editor 39
Appendix A: Creating storyboards 41
Appendix B: MeteoCAL XML 45
Appendix C: Installing MeteoCAL directory structure and tools 57
Appendix D: Common errors when processing XML files and their causes 59
Appendix E: Localisation – Adding a new language 60
Appendix F: Localisation – Setting which languages will be displayed 62
Appendix G: General Internationalisation/Localisation Guidelines 63

This manual is intended to be used for many different tasks. To help you find the correct section for what you want to do the following are typical tasks and they highlight relevant sections of the manual.

Table 1: Recommended reading for common tasks

Task:	Read sections:
Maintain existing material on the EUMSAT MSG-CAL web server.	1 and 2[1]
Change the structure of existing material (on EUMETSAT server or elsewhere)	1 and 5
Create a new unit-of-study composed of several knowledge objects and prepare it for web or CD-ROM presentation based on existing templates.	1, 4 and 7
Create a new module by combining existing units of study.	1 and 3
Set up a web server or an intranet directory structure so that it can provide MeteoCAL modules.	1, 6, 7 and 8
Develop new templates or create knowledge objects from scratch.	9 and 4
Prepare to add modules in a new language	Appendices E and F

MeteoCAL is designed to operate on PC systems with Windows (98, 2000, NT) running browsers Netscape 6.2+ and Internet Explorer 5.5+ on screens with a resolution of at least 1024x768.

IMPORTANT: Before making any substantial changes to MeteoCAL modules or units-of-study, please ensure that a copy has been made of the old version so that you can revert back to it if you are not happy with your changes. The simplest way to achieve this is to carry out a complete daily back-up of the source directory. Note that it is the **source** directory that needs to be backed up. If you have the **source** you can always reproduce the **www** or **cdrom** versions of MeteoCAL.

All sections of this manual assume that the reader is familiar with the experience of working through EUMETSAT's MSG-CAL modules on the web or on CD-ROM.

[1] Note that Chapter 2 refers to the MSGMaintain tool which only exists on the EUMETSAT server and is not a part of the MeteoCAL tools.

1: The Structure of MeteoCAL modules

A MeteoCAL module is a collection of units-of-study to which structure has been applied to offer the student one or more routes through the collection. Some parts of the pathway may be essential, in which case the student cannot avoid them, but in other parts several alternative paths might be offered on the basis of a student's interest, or past experience, or available time.

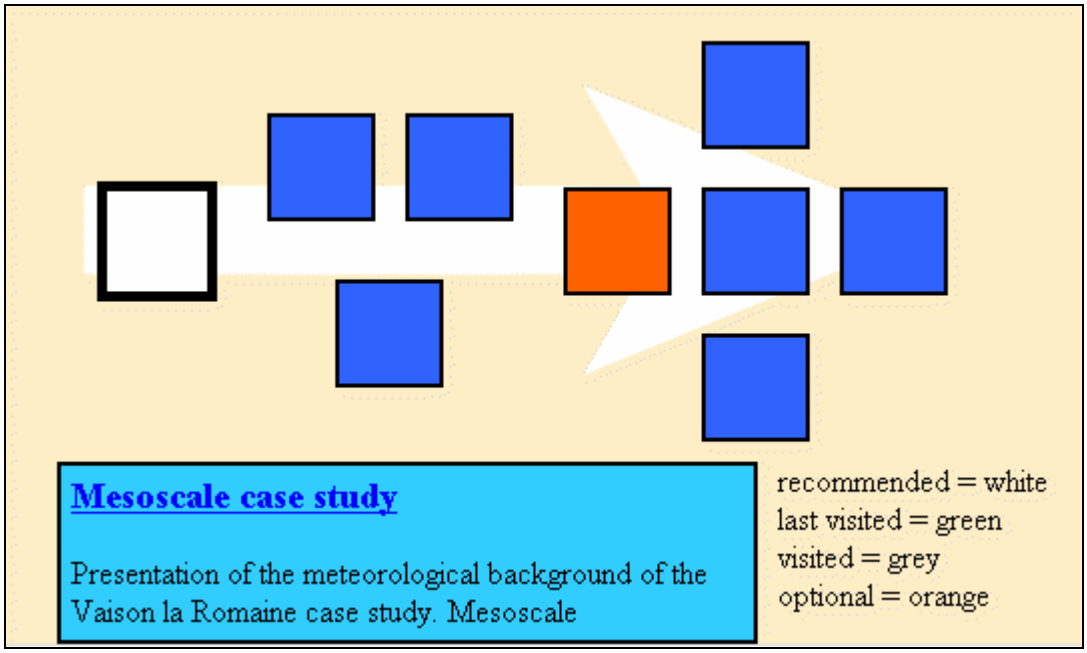

Mesoscale case study
Presentation of the meteorological background of the Vaison la Romaine case study. Mesoscale
recommended = white
last visited = green
visited = grey
optional = orange

Figure 1: Example of a screen view of navigation through a module. Each unit-of-study is represented by a box. The student is expected to progress from left to right but vertical columns reflect options at that stage. A small pop-up window appears when the mouse is placed over each box to describe what is contained in that unit-of-study. A student may choose to take any unit-of-study simply by clicking on the box. The units-of-study which are recommended as the next stage are shown in white, those already visited in grey and the last module visited is green. Optional modules offering non-core information are shown in orange.

Each unit-of-study is composed of several web pages, usually only a few (2-4 pages). Within these web pages there are objectives, interactions, images and text. So one unit of study may have several pages, images, and interactions associated with it.

In addition, a unit-of-study may include references to several items in the reference library and several items which are on other web sites (external URLs). These are not directly part of the unit-of-study but are made available when the unit-of-study is viewed in its "environment". They provide supporting material to give added depth to the learning experience.

Table 2: Hierarchical structure and the support for learning that exists at each level.

Hierarchical level	Scope
Course	A course is an instructional unit usually defined by an organisation or a teacher. The scope of a course is beyond this document because any teacher or organisation is free to use MeteoCAL modules in any type of course. Typically a course could include some classroom instruction, which might include demonstrations based on parts of the MeteoCAL. Or it might include some self-study in a fixed timetable using MeteoCAL modules. It could also be used as continuous professional development

	(CPD) in which small elements, for example individual units-of-study, could be taken in the workplace according to time or knowledge demands. MeteoCAL is designed to be flexible in its uses and courses in which MeteoCAL is used will take many different forms.
Module[2]	A module is a component of a course which has specific learning objectives and specific knowledge/expertise pre-requisites. MSG-CAL modules have been initially designed to take about 45 minutes of effort by a student working in self-study mode, but since there is usually more than one route through a module it could take significantly more or less time. A module contains a list of the units-of-study that it contains, navigational information which defines the possible routes through these units-of-study, and explanatory notes which form the "glue" between the units of study. Since units-of-study are designed to be reused in many different contexts they are quite independent from each other. It is through the explanatory notes in the modules that linkages between units-of-study in any particular module can be created. Thus phrases such as "having seen the application of this channel in fog detection you can now go on to see how this information fits into NWP products" can be used at this level although they cannot be used in individual units of study as they would restrict the context.
Unit-of-study	A unit-of study also has learning objectives and pre-requisite knowledge. This guides anyone who is creating a module to indicate what information the student needs. Note that the student does not need all pre-requisite information in the form of other units-of-study - it could be provided through any other form of instruction in a course or even from past experience. A unit of study includes one or more knowledge objects and an environment supporting these knowledge objects. Each knowledge object is equivalent to a web page and it is good practice to keep the number of knowledge objects in a unit-of-study small (typically 2 to 4). This helps to ensure that the unit-of-study can be reused in other contexts. The environment which support each unit-of-study comprises three elements: • Reference library: the reference library contains all documents which support MeteoCAL units-of-study but for each specific unit-of-study a list is provided of the most relevant items in the reference library. Thus the student is presented with the documents in the reference library (Word, PowerPoint, pdf, diagrams and tables) which are most relevant to the current unit-of-study. • Communication modes: the MeteoCAL environment has the potential to support links to email, discussion groups and synchronous chat facilities. The provision of these facilities is not included since there are many available sources. Email support gives the opportunity to nominate a general technical support person and also a specialist tutor for the topic of the particular unit-of-study. • External URLs: In order to maintain focus within a unit-of-study it is best not to include links to external sites. However, a list of relevant links can be provided to the student through this environment. These are external web sites with direct relevance to the topic of the unit-of-study. Students can divert to one of these web sites at any time but they are opened in another window to ensure the student does not lose their place in the unit-of-study.
Knowledge Object	A knowledge object is equivalent to a web page. It differs from a web page because it does not specify the style or mode of delivery of the knowledge. A typical web page will include information and zero or one interactions (see the "interactions"

[2] Strictly speaking a "module" is also a unit-of-study which only contains other units-of-study, but we will continue to use the term module to describe the collection with associated navigational structure and unit-of-study to describe the component blocks.

	table below). Knowledge objects are reusable in different units-of-study in the same way that units-of-study are reusable in different modules.

In order to make knowledge objects as interactive as possible a set of generic interactions is provided which can be reused in many different situations. These are listed in the table below and tem[plates are provided for knowledge objects containing each of these interactions. Further details are given in section 4.

Table 3: "Standard" interactions for which templates are provided in the MeteoCAL tools. Other interactions can also be built using the XML entities described in detail in the appendix.

Interaction name	Description
Multiple choice question	One or more questions in which several choices are presented, one of which is the correct answer. Both the question and the feedback for correct or incorrect answers may contain images. Questions are displayed one after another.
Matrix question	A series of questions in which the potential answers are all of the same type and therefore presented in columns. A very simple example is where the answers are either "true" or "false" so there are two columns with these headings. However, the number of choices is not limited to two. Both the question and the feedback for correct or incorrect answers may contain images.
Image hotspot	An image is presented with a question whose solution is found by clicking on some location in the image. There may be several correct and incorrect areas in the image. Each area can be independently defined in the shape of a rectangle, circle or polygon. Images may be multi-spectral in which case the images are presented in a stack which can be viewed one image at a time. For multi-spectral images the answer areas exist on all images. If several images are to be displayed simultaneously as part of the question they should be formed into a single composite image. Feedback for correct and incorrect answers may also include images.
Text drag and drop	The question may include a background image and the interaction involves moving one or more text phrases to specified locations (for example to form a list, or label a diagram). One draggable object may be located in multiple droppable locations. Feedback for correct and incorrect answers may also include images.
Image drag and drop	Similar to text drag and drop but the draggable objects are images.
Animation	A series of .gif, png or .jpg images are displayed and played as an animation. The controls available to the student include: start, stop, pause, step forward, step backward, play once, loop, swing, increase speed, decrease speed, show only every second image (to demonstrate different between first and second generation Meteosat sampling periods).
Slide show	A series of .gif or .jpg images is presented each with a text caption. The student can step backwards and forwards through the sequence.
Slide show combined with MCQ	Similar to Slide show but for each image in the sequence a multiple choice question can optionally be presented.
Special	Any valid code on a web page can be included using this option which makes it easy, for example, to include Java applets.

All of these interactions are intended for formative assessment in which the student gets immediate feedback to aid the learning process. The feedback box can be left on screen while trying different answers to explore the different arguments. In all but drag-and-drop questions the feedback area is draggable to a convenient part of the user's screen. In the mcq’s, matrix mcq’s and drag-and-drops you may use generic feedback for correct or incorrect answers.

To create knowledge objects based on one of these generic interactions it is simplest to use the templates for these knowledge objects described in section 4.

2: Changing an existing Unit-of-Study

To change an existing unit-of-study you can use the web-based maintenance tool. This tool is available at http://www.intrallect.com:8080/msgmaintain2/ and is only available for the maintainance of the MSG-CAL modules on the EUMETSAT web site.

Normally you will already know what you want to change when you start to use this tool. The information you need to know is the language (English or French), the module, the unit of study and the knowledge object within the unit of study. All but the last of these will probably be very obvious but the last may not be so clear. Imagine you spot an error on a page and you are not sure which knowledge object it is. By right-clicking on the page containing the error you will be able to select "View frame info..." (Netscape) or "Properties" (IE) and take a note of the URL of the problem knowledge object, for example http://www.intrallect.com/devel/msgcal/www/english/msg/rds/5b/usefprod.htm.

In the **msgmaintain** tool you can navigate up and down the file structure using the file browser shown below:

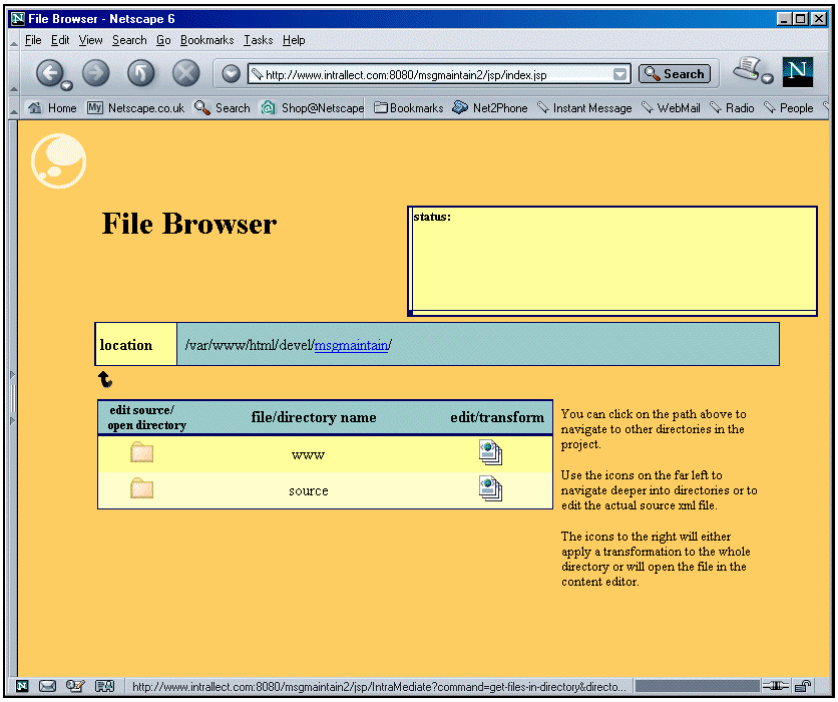

File Browser - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
http://www.intrallect.com:8080/msgmaintain2/jsp/index.jsp
Search
Home Netscape.co.uk Search Shop@Netscape Bookmarks Net2Phone Instant Message WebMail Radio People
File Browser
status:
location
/var/www/html/devel/msgmaintain/
edit source/ open directory
file/directory name
edit/transform
www
source
You can click on the path above to navigate to other directories in the project.
Use the icons on the far left to navigate deeper into directories or to edit the actual source xml file.
The icons to the right will either apply a transformation to the whole directory or will open the file in the content editor.
http://www.intrallect.com:8080/msgmaintain2/jsp/IntraMediate?command=get-files-in-directory&directo...

Figure 2: Top level view of the file browser in the msgmaintain tool.

Since you need to edit the source you will choose *source*.

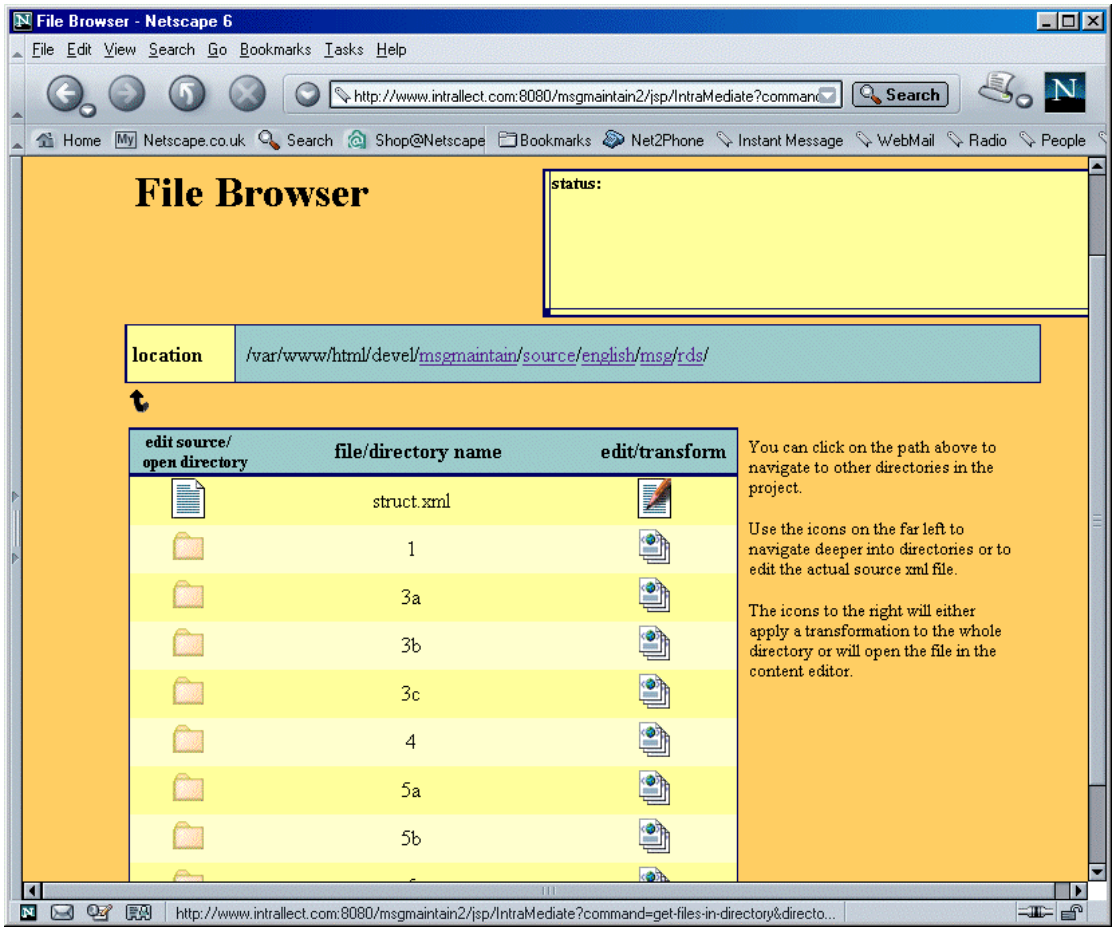

File Browser - Netscape 6
File Edit View Search Go Bookmarks Tasks Help
Search
Home Netscape.co.uk Search Shop@Netscape Bookmarks Net2Phone Instant Message WebMail Radio People
File Browser
status:
location
/var/www/html/devel/msgmaintain/source/english/msg/rds/
edit source/ open directory
file/directory name
edit/transform
struct.xml
1
3a
3b
3c
4
5a
5b
You can click on the path above to navigate to other directories in the project.
Use the icons on the far left to navigate deeper into directories or to edit the actual source xml file.
The icons to the right will either apply a transformation to the whole directory or will open the file in the content editor.

Figure 3: View of a module showing the struct.xml file which contains the navigation information and one directory for each of the units-of-study. Click on the folder on the left to enter the unit-of-study you wish to modify. In the example given in the text the directory to select would be 5b. Note that the symbol to the right of the directory names allows you to transform a whole directory, and all its sub-directories, with a single mouse click.

Once you have identified the file you need to edit (in the example above the file will be source/english/msg/rds/5b/usefprod.xml - note that the file on the web has a .htm extension but the source file has a .xml extension) you have the choice of several ways of editing it.

1. Edit any/all text in the entire knowledge object. Simple and fast, particularly if changes are needed in more than one part of the knowledge object.
2. Edit the text in one particular part of the knowledge object. This is best if you are not sure of the structure of the knowledge object as you can edit different parts separately.
3. Edit the .xml file directly. This is fast but not recommended unless you are already familiar with XML as it is easy to create an invalid file.

Each of these methods is now described in more detail:

Edit any/all text in the entire knowledge object

Click once with the left mouse button on the file name (e.g. usefprod.xml). A new window appears in which all the editable text elements have been extracted into separate text boxes.

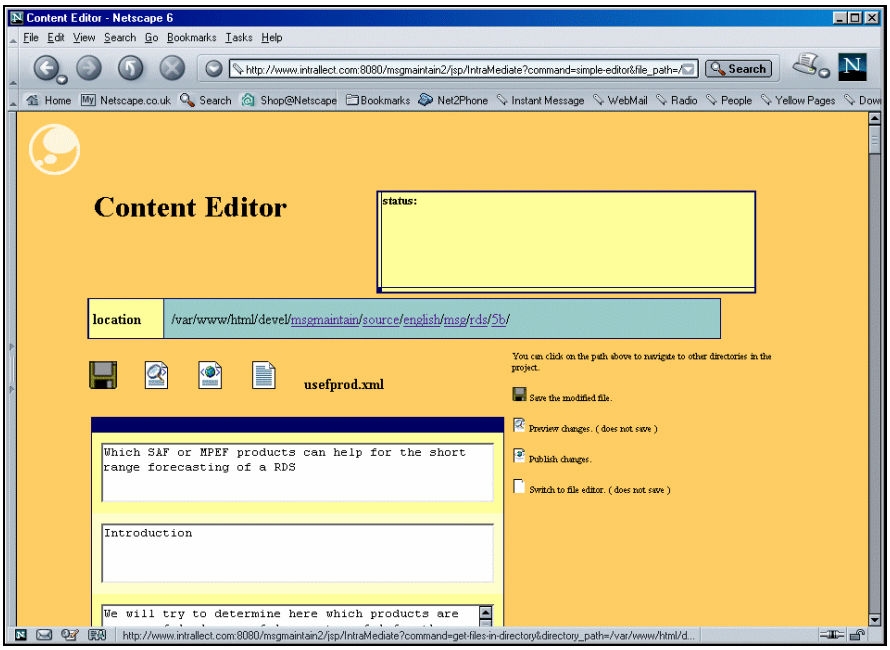

Content Editor - Netscape 6
Content Editor
status:
location
/var/www/html/devel/msgmaintain/source/english/msg/rds/5b/
usefprod.xml
You can click on the path above to navigate to other directories in the project.
Save the modified file.
Preview changes. (does not save)
Publish changes.
Switch to file editor. (does not save)
Which SAF or MPEF products can help for the short range forecasting of a RDS
Introduction
We will try to determine here which products are

Figure 4: The content editor view of the file usefprod.xml with each of the text elements on the page displayed in a text box and available for editing.

Change any of the text as you wish. You can change many parts at the same time. Once you have finished you can use the preview, publish or save buttons as described below.

Edit the text in one particular part of the knowledge object
Click first on the file you want to edit. Then click on the preview button and a window will appear in which there are several Edit tags.

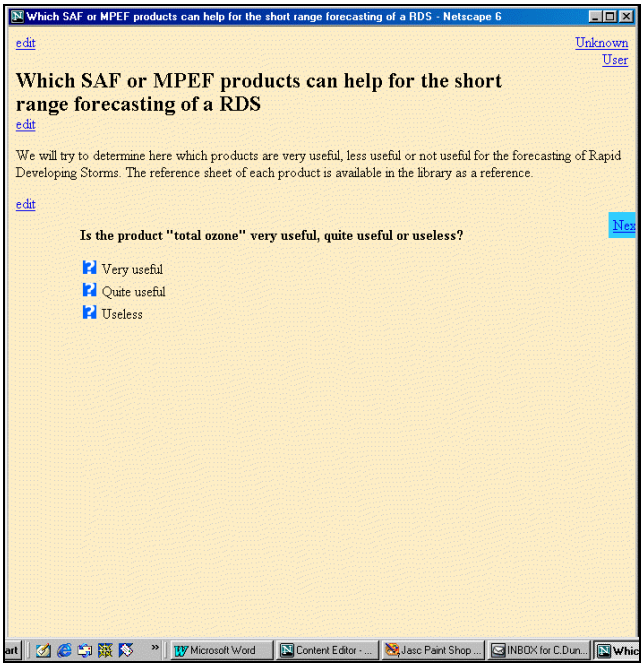

Which SAF or MPEF products can help for the short range forecasting of a RDS - Netscape 6
edit
Unknown User
Which SAF or MPEF products can help for the short range forecasting of a RDS
edit
We will try to determine here which products are very useful, less useful or not useful for the forecasting of Rapid Developing Storms. The reference sheet of each product is available in the library as a reference.
edit
Is the product "total ozone" very useful, quite useful or useless?
Very useful
Quite useful
Useless

Figure 5: Preview window with edit tags which show which sections of the page are editable.

In the example in Figure 5 the three Edit tags correspond to the title, the paragraph and the question.

Next, dentify which section of the knowledge object you wish to edit and click on the appropriate Edit tag. If the section you are editing is simple, for example a title, paragraph or caption, then you will simply see that text in a text box and you can edit it.

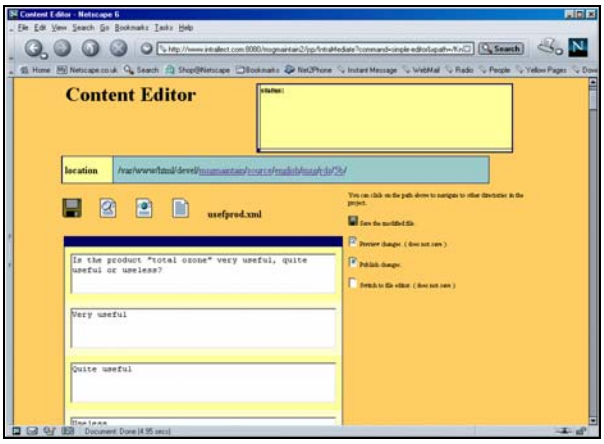


Figure 6: Content editor view of a question selected from the preview window as shown in figure 5. The question, options and feedback can all be edited.

If it is more complex, for example a multiple choice question, you will be presented with all the editable text in separate text boxes.

In the example in Figure 6 you can see the question, the displayed options, and the feedback provided when each option is selected would be accessible by scrolling the window down. Note that this feedback is only visible when a student clicks on one of the options.

Edit the .xml file directly

To edit the XML file directly click on edit source in the msgmaintain file browser window. The entire file will be presented in a text box for editing as shown in Figure 7. In some cases the file may not look as well-formatted and readable as this example but the appearance depends on the editor used when the file was last edited and cannot be guaranteed.

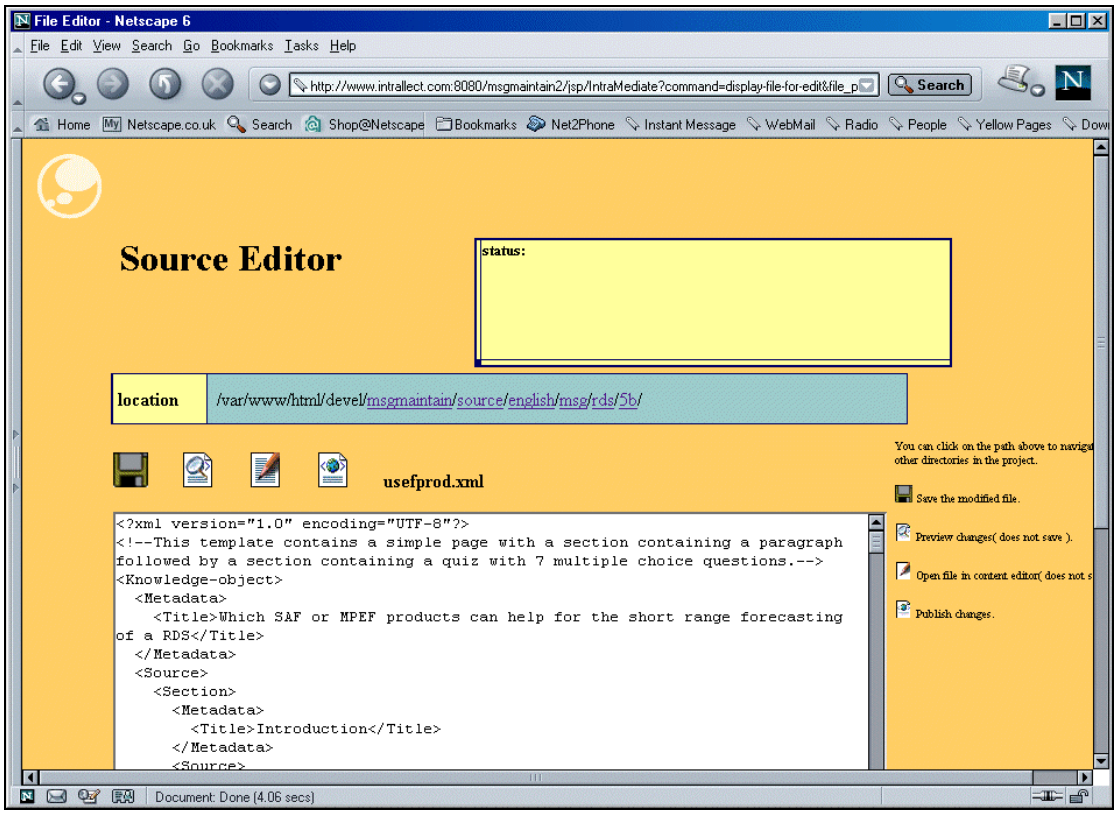


Figure 7: Source editor view of the file usefprod.xml. This displays the XML source which can be edited. Care must be taken to ensure that the file still contains valid XML after editing.

Publishing and saving
On all of the editor windows there is a set of buttons for previewing, publishing and saving.

Figure 8: From left to right these icons offer facilties for : save (to save the changes without transforming the file to publish the results to the web); preview (to see the the web-view of the edited file. The file can be previewed without being saved); open (opens the file in the content editor); publish (saves the file, transforms it and publishes the changes on the web).

The preview button creates the window in which the web page is shown with Edit tags described above and shown in Figure 5. This page can be used to check that the changes you made are working correctly before you publish or save them.

The save button is used to save the changes in the source file but no transformation of the source file to the web and cdrom is carried out. This option might be used when a large number of files are all to be change at the same time and you want to publish all the changed files at the same time. In that case you save all the files and then go to the file browser. Use the file browser to find a directory level which contains all the files you have changed and select transform dir which will transform all the XML files in the directory and all its subdirectories. This transformation will cause all these files to be published in both www and cdrom versions in one action. This transformation cannot be undone. Any mistakes made at this stage will need to be rectified from your daily backup.

The publish button is used to cause a single file to be transformed into its www and cdrom versions. This happens immediately you press publish. The source file is automatically save when you press publish.

Final output
Once a new file has been published you can view the changed file as it would be viewed in the www directory by checking it at the usual web address for the EUMETSAT MSG-CAL site.

3: Creating a New MeteoCAL Module

MeteoCAL modules are based on a way of describing learning activities in XML. The specific method used is based on EML (Educational Modelling Language) originally produced by the Open University of the Netherlands. (See Koper (2001)[3] for more information.) EML is based on many years of research and is capable of supporting many different pedagogical styles. The value of EML was recognised by the e-learning specification group IMS[4] when it began to develop a *learning design* specification. The core elements of EML have been adapted for the IMS Learning Design specification and they have incorporated some of the other IMS specifications such as Metadata and Content Packaging. MeteoCAL is based on a specific interpretation of EML which pays attention to the direction in which IMS Learning Design (compliance level A) is moving. However, the deadline for MeteoCAL completion is considerably earlier than any conclusion on the IMS Learning Design specification. As a result MeteoCAL is an implementation based on the principles of both EML and IMS Learning Design, but it cannot be guaranteed to comply with the final version of these specifications. Since no *delivery system* for IMS Learning Design will exist until after the specification is finalised, Intrallect has produced a delivery system capable of supporting the MeteoCAL modules. Modules produced following the guidelines contained in this manual will be supported by the MeteoCAL delivery system.

Figure 9: Top-level structure of a MeteoCAL module or unit-of-study. Although the same basic structure is used for both modules and units-of-study (even to the extent of using the element <Unit-of-study> as the root element in both) the information contained in each is quite different.

The building block used to create MeteoCAL is a *unit-of-study*. Each unit of study usually has six components. *Metadata* is data which describes the content and intended audience for this unit-of-study. *Roles*, in the case of MeteoCAL is very simple as it is a single learner. The *learning objectives* describe what is to be learned and the *prerequisites* define what should be known before starting. The *content* is the material which will be provided in this unit-of-study and the *method* describes the way in which it will be provided.

Note that this is simply a building block so it is also possible to define another unit-of-study which contains several units-of-study. In fact this is how an MeteoCAL module is produced. A module is a substantial item for learning, typically occupying a learner for about 45 minutes. A single module will contain several units-of-study. This allows substantial flexibility. For example a module may contain a theoretical part followed by several case studies. The theoretical part and case studies may each be units-of-study. One teacher might want to remove the theoretical part completely without replacing it. Another teacher might want to replace one case study with another in a more appropriate geographical region. The units-of-study inside a module are intended to be "swappable" parts which can be added or removed to customise the lesson. A module is itself a unit-of-study although it has a slightly special structure.

[3] Koper, R, 2001, "Modelling Units of Study from a Pedagogical Perspective" (http://eml.ou.nl/introduction/articles.htm
[4] http://www.imsproject.org/

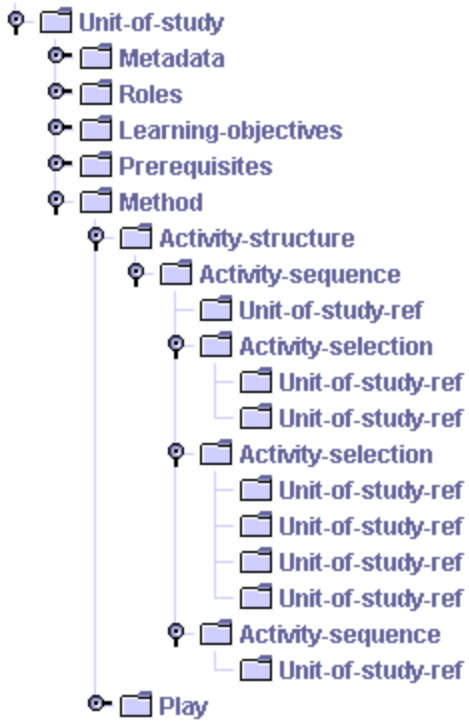


Figure 10: Top-level structure of a module expanded to show a typical activity-structure which describes the order in which the compnent units-of-study can be taken.

A module is a unit-of-study with almost no content. The only content the module file contains is a description of each unit-of-study which acts to show its relevance in this module and a few paragraphs describing the overall purpose of the module. The method includes an activity-structure. *The activity-structure* includes several activities some of which are described as *activity-sequences* while others are described as *activity-selections*. An activity-sequence is used when one activity is to be followed by another. An activity-selection is used when several options are to be presented and the route through these selections depends on the user's own decisions. Within the activity-sequences or activity-selections are a series of unit-of-study-refs. These are simply references to units-of-study which appear at that point. The structure shown in Figure 10 is equivalent to a module composed of eight units of study. The first one must be taken first (presumably an introduction), the next two may be taken in either order, or only one may be taken. Then four more options are presented. At least one of these four options must be taken although two or three or four may be taken, before the final unit-of-study (presumably the conclusions, or perhaps an assessment) is presented. The way that this set of options would appear to the student is shown in Figure 11.

The *play* item simply relates the roles to the activities so in this case it links the learner to the activity structure (although this detail is not shown in the diagram).

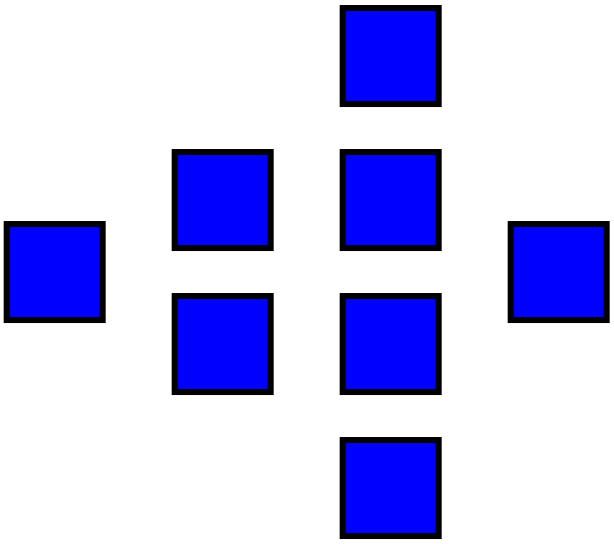


Figure 11: Sample of the navigation view of an activity structure as defined in Figure 10.

The remainder of this section describes how to create a "module description file" and the appropriate directory structure. In the source part if the directory structure (see section 6 if you are not already familiar with the directory structure) identify the module level. In source\english\msg you will find the calib, rds and channels modules of MSG-CAL. In this example we will add a new module alongside these. This new module will be referred to as new-module but you can call it whatever you like (just remember to use your name wherever new-module is referred to).

Create a directory called new-module. In this directory the module description file should be called struct.xml. You can create that file from scratch if you choose (see section 8 on choosing and using an XML editor) but the simplest method is to use an existing struct.xml file as a template. Before attempting this you may find it useful to create a storyboard for the module (see Appendix A). You can copy the struct.xml file from any existing module. In this example we will assume that you have copied the file source\english\msg\channels\struct.xml and placed it in the new-module directory. The file is shown below split into sections with a commentary on how to edit each section. To edit this file use your preferred editor or word processor, e..g. notepad wordpad or Word. You must remember that there should be no formatting in these .xml files so when you save the files always save them as text files not as Word documents or .rtf files.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

This first line should always be exactly in this form, do not edit it.

```
<Unit-of-study>
<Title>1.6 and 3.9  micrometer  channels on MSG (Meteosat Second
Generation)</Title>
</Metadata>
```

Add the title of the module as shown in bold text. Do not edit the parts inside <>

```
<Roles>
<Learner Id="learner" Max-persons="1" Min-persons="1"/>
</Roles>
```

Leave this section as it is.

```
<Learning-objectives>
<Learning-objective Id="o1">
<Objective-description>
<P>Develop a basic understanding of the signals received in the
new 1.6 and 3.9  micrometer  channels</P>
</Objective-description>
<Objective-type>
<Insight/>
</Objective-type>
```

You may choose to change all the parts shown in bold. The Id o1 is suitable for the first learning objective and can be left unchanged. The description of the learning objective should be set to whatever you have decided – perhaps you have already defined it in your storyboard, in which case you can cut and past it into the struct.xml file.

```
</Learning-objective>
<Learning-objective Id="o2">
<Objective-description>
<P>Become familiar with the meteorological phenomena which can
be identified with the new 1.6 and 3.9  micrometer  channels on MSG</P>
</Objective-description>
<Objective-type>
<Knowledge/>
</Objective-type>
</Learning-objective>
```

Example of a second learning objective. If you do not have a second learning objective then delete this section. If you have more learning objectives then duplicate this section.

```
</Learning-objectives>
```

This statement closes the learning objectives section.

```
<Prerequisites>
<Prerequisite Id="p1">
<Prerequisite-description>
<P>A good knowledge of the use of Meteosat (MOP) channels</P>
</Prerequisite-description>
<Prerequisite-type>
<Knowledge/>
</Prerequisite-type>
```

Prerequisites are treated in an almost identical way to learning objectives.

```
</Prerequisite>
<Prerequisite Id="p2">
<Prerequisite-description>
<P>An awareness of the requirements of forecasting and nowcasting</P>
</Prerequisite-description>
<Prerequisite-type>
<Situational-factor/>
</Prerequisite-type>
</Prerequisite>
```

Example of a second prerequisite. If you do not have a second prerequisite then delete this section. If you have more prerequisitess then duplicate this section.

```
</Prerequisites>
```

Closes the prerequisite section.

```
<Content>
<Environment>
<Knowledge-object>
<Metadata/>
```

None of the above needs any changes. Note that <Metadata/> is empty and is not strictly required.

```
<Source>
<P>This module offers an introduction to the two channels
which are new to Meteosat with the Second Generation.</P>
<P>You can use the module as a complete survey or simply as a
way of sampling some aspects of these new channels.</P>
<P>The module starts with a small example and ends with a quiz
to find out what you have learned, Between these are two
sections. The first introduces the two channels at 1.6 and
3.9  micrometer . You can look at either or both of these. The next
stage discusses four different applications of these
different channels to show their importance.</P>
<P>The order in which you take the various units in this
module is entirely up to you, but you may find it helpful to
start at the left and work your way to the right on the
diagram below.</P>
<P>Credits: Thanks to Scott Bachmeier, SSEC, University of
Wisconsin for advice and support in the preparation of this module.</P>
</Source>
</Knowledge-object>
</Environment>
</Content>
```

This text is a description of the module that will be displayed at the start of the module and every time the student leaves one unit-of-study and is about to enter another. It should describe the context of the units of study. It is also a good place to give module-level credits. You can have as few or as many paragraphs as you like. Note that every paragraph must be enclosed between <P> and </P>.

```
<Method>
<Activity-structure Default-visibility="Show" Id="module">
<Activity-sequence Link-name="module">
```

Start of the navigation definition. These statements should be left unchanged. By default one or more units-of-study must form a sequence.

```
<Unit-of-study-ref Link-name="Motivation" Ref-worldwide-unique-
id="motiv/struct.xml"><?popup Why are the new 1.6 and 3.9  micrometer
channels important? This unit introduces how these channels differ from
simple visible, water vapour and thermal IR channels.?></Unit-of-study-
ref>
```

Three aspicts of the first unbit of study are defined here. The Link-name is the text that will appear as a title for this unit-of-study. The Ref-worldwide-unique-id is the location of the struct.xml file for this unit-of-

study relative to the struct.xml file for the module which you are editing. This should be in a subdirectory named for the unit-of-study, in this example it is called motiv. Popup is the text that will appear to describe this unit of study to the student when he/she is given the option of choosing the next unit of study

```
<Activity-selection Number-to-select="2">
```

Identifies the next two unit-of-study-refs as choices rather than a linear sequence.

```
<Unit-of-study-ref Link-name="1.6 micrometre channel" Ref-worldwide-
unique-id="1-6/struct.xml"><?popup Introducing the capabilities of the
new 1.6 micrometer  channel. If you do not know about the properties of
this channel you should work through this unit before looking at the
applications of this channel.?></Unit-of-study-ref>

<Unit-of-study-ref Link-name="3.9 micrometre channel" Ref-worldwide-
unique-id="3-9/struct.xml"><?popup Describes the properties of the new
3.9 micrometer  channel. Work through this unit before looking at the
applications unless you are already familair with 3.9  micrometer
channels from other instruments scuh as AVHRR or MODIS.?></Unit-of-
study-ref>
```

Notrice that newlines have no meaning in these XML files. Inlcude then to make it more readable if you wish but they are not significant.

```
</Activity-selection>
```

End of the selection of two options.

```
<Activity-selection Number-to-select="4">
```

Start of a new selection of four options.

```
<Unit-of-study-ref Link-name="Snow-cloud discrimination" Ref-worldwide-
unique-id="snow-cl/struct.xml"><?popup Distinguishing between cloud and
snow is difficult using visible or thermal IR channels but the new 1.6
micrometer  channel makes discrimination between them much easier. It
is best to check the properties of the 1.6  micrometer  channel first
if you haven't already done that.?></Unit-of-study-ref>

<Unit-of-study-ref Link-name="Cloud phase determination" Ref-worldwide-
unique-id="cloud-ph/struct.xml"><?popup Using the 1.6  micrometer
channel makes it much easier to tell the difference between water
clouds an ice clouds. It is best to work through the 1.6  micrometer
module before looking at this application.?></Unit-of-study-ref>

<Unit-of-study-ref Link-name="Fog detection" Ref-worldwide-unique-
id="fog/struct.xml"><?popup Detection of fog is imporved on MSG using
both the 3.9 micrometer  channel and improved scan rate. You should
know about the properties of the 3.9  micrometer  channel before
working on this unit.?></Unit-of-study-ref>

<Unit-of-study-ref Link-name="Fire detection" Ref-worldwide-unique-
id="fire/struct.xml"><?popup Detection of wild-fires is easier using
the 3.9  micrometer  channel. This unit describes using hotspots
(detected using the 3.9 micrometer  channel) and smoke plumes to
identify fires.?></Unit-of-study-ref>
</Activity-selection>
```

End of selection of four units of study.

```
<Unit-of-study-ref Link-name="Conclusions" Ref-worldwide-unique-
id="concl/struct.xml"><?popup Test yourself on what you have learned.
Or if you prefer try this quiz first, and see what you still needd to
learn, then work through the module and see if you have improved by
taking the test again.?></Unit-of-study-ref>
```

Final unit of study in the sequence.

```
</Activity-sequence>
</Activity-structure>
```

End of the overall sequence and of the structure for navigation.

```
<Play><!--This is a top-level "navigation-only" UoS so only contains
play details for other UoSs--><Role-ref Id-ref="learner"/>
<Activity-structure-ref Id-ref="module"/>
</Play>
```

Simply defines the module is to be played for the learner. This section should not be modified. Notice that comments can be included if placed between <!--and -->.

```
</Method>
```

End of navigation section.

```
</Unit-of-study>
```

End of module definition file.

```
<!-- Keep this comment at the end of the file
Local variables:
mode: xml
sgml-omittag:nil
sgml-shorttag:nil
sgml-namecase-general:nil
sgml-general-insert-case:lower
sgml-minimize-attributes:nil
sgml-always-quote-attributes:t
sgml-indent-step:2
sgml-indent-data:t
sgml-parent-document:nil
sgml-exposed-tags:nil
sgml-local-catalogs:nil
sgml-local-ecat-files:nil
End:
-->
```

Keep this in the file and do not modify

This completes the struct.xml file for the module structure. Save the file – remembering tio ensure it is saved as a text file and not a formatted file such as .rtf or .doc. In order to process the struct.xml file see the section on Transforming Source Files. However it is probably best to wait unti you have units of stidy in place before transforming struct.xml files as you cannot easily view the output without units of study.

4: Creating a New MeteoCAL Unit-of-Study

A unit-of-study is the basic educational unit used in MeteoCAL. It is the smallest unit that will deliver learning measured against a learning objective. A unit-of-study may contain several resources in different forms. These require the structure of a unit-of-study to enable them to achieve the learning objectives. However, the resources used in a unit-of-study may be reused in other contexts. A module is a substantial set of material composed of several units-of-study in a flexible sequence that makes it adaptable to different forms of learning.

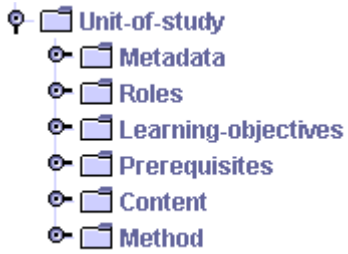


Figure 12: Basic structure of a unit-of-study. Note that this structure is identical to that of a "module" which is also technically a unit of study. However, the information stored at the unit of study level is quite different.

As a guide, a unit-of-study should normally be equivalent to a few web pages, perhaps from 1 to 4 pages. Since it is so small, there is no complex structure and all pages in a unit-of-study are assumed to be sequential. Each page is described as a knowledge -object. A MeteoCAL module is a special unit-of-study which contains several of these basic units-of-study. However, a module has a different purpose which is described in "Creating a new MeteoCAL module".

The structure of any unit-of-study is as shown in Figure 12. In MeteoCAL module description files the emphasis was on the *method* part which defines the navigation through the module. Here the emphasis is on the *content* part which includes both the *knowledge-objects* and the *environment*. In both cases metadata is used only to hold the title of the unit-of-study (it is recommended that full IEEE LOM metadata for each unit of study is kept separately from the unit-of-study.) Also roles are always the same for MeteoCAL involving only a single learner. In order to differentiate between a unit of study used to define a module and a basic unit-of-study consisting of a few web pages we use the terminology "module description file" for the module and unit-of-study is used only to mean the basic building block of a few web pages with its associated environment.

In a unit-of-study, learning-objectives and prerequisites are included for information but are not displayed to the learner. Their purpose is to allow a module designer to see what the unit-of-study is intended to achieve and what prior knowledge must be provided. As discussed above, the main emphasis in the unit-of-study is the content section.

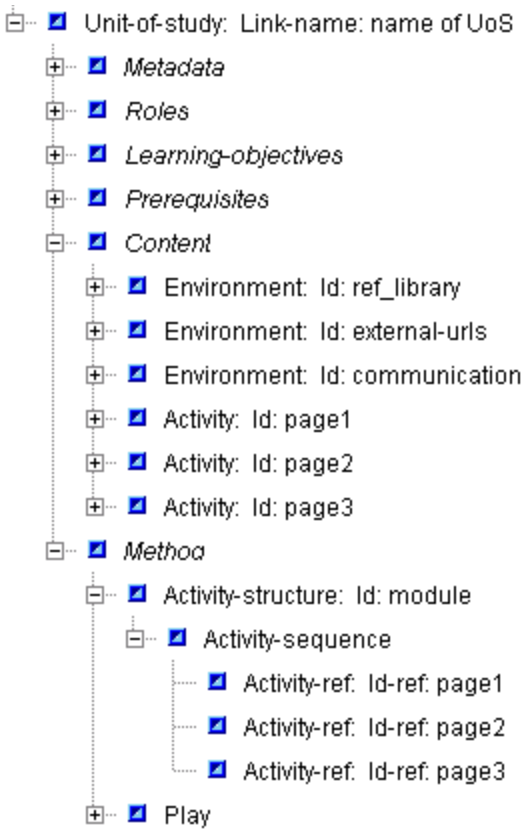


Figure 13: Expanded view of a unit-of-study showing the support environments and linear sequence of activities. Each activity contains one knowledge object..

The content section contains two main parts - the environment and the activities, Figure 13. The environment defines the services which are available while the learner is working on the unit-of-study. In MeteoCAL most of this environment is fixed but some components of it can change from one unit-of-study to the next. It is these components that are configured here: the reference library, the external URL links, and communications support (although a communications support section has been included it is acted upon in MeteoCAL transformation because the MSG-CAL communications used only a fixed email address.)

The activity part contains the pages which make up this unit-of-study. Add or remove activities so that they reflect the number of pages to be included in any particular unit-of-study. The method part shows that these pages are delivered in sequence. This should not be modified .

An author of a new unit-of-study will need to configure both the environment and the activity parts. This can best be achieved by working from an existing example. The unit of study description is contained in a struct.xml file in the unit of study subdirectory. To create a new unit of study create a new subdirectory below the module in which the unit of study is to be used. Then copy and existing struct.xml file from one of the MSG units of study, say the channels/fire unit of study. That file can be edited using any familiar editor or word processor. The following example shows what should be changed in the channels/fire/struct.xml example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Do not change the first line.

```
<Unit-of-study Link-name="fire">
```

The name of the unit=of study should be set as Link-name.

```
<Metadata>
<Title/>
<Description/>
</Metadata>
```

This metadata section is empty and need not be included.

```
<Roles>
<Learner Id="learner" Max-persons="1" Min-persons="1"/>
<Staff Id="manager"/>
</Roles>
```

Should not be changed – roles are always the same in MeteoCAL modules.

```
<Learning-objectives>
<Learning-objective>
<Objective-description>
<P>To become familiar with the use of MSG for fire detection</P>
</Objective-description>
<Objective-type>
<Knowledge/>
</Objective-type>
</Learning-objective>
</Learning-objectives>
```

Change the learning objective description and type to your own choice. A unit of study should not have more than a single learning objective. If there are two learning objectives then there should be two units of study.

```
<Prerequisites>
<Prerequisite>
<Prerequisite-description>
<P>A basic understanding of the properties of the 1.6 and 3.9
micron channels on MSG</P>
</Prerequisite-description>
<Prerequisite-type>
<Knowledge/>
</Prerequisite-type>
</Prerequisite>
</Prerequisites>
```

Change the parts in bold to reflect the prerequisite knowledge. If more than one prerequisite is required then duplicate all the lines between <Prerequisite> and </Prerequisite> inclusive.

```
<Content>
```

Beginning of content section.

```
<Environment Id="ref_library">
```

Beginning of Reference library section. No modification required. The reference library contains documents that give the student addition information for more in-depth learning.

```
<Knowledge-object Type="ref-lib">
<Metadata>
<Title>Not used</Title>
</Metadata>
<Internet-source
Link-name="Introduction to potential fire applications of MSG"
Url="../../library/fireintr.doc"/>
</Knowledge-object>
```

Definition of the first object in the reference library for this unit of study. Link-name is the title that the student will see for this object. URL is the relative path name to the document. Since all reference library documents are held in the same place for all units of study care should be taken to ensure that they all have unique names. Note that the parts between <Metadata> and </Metadata> are not used and could be omitted.

```
<Knowledge-object Type="ref-lib">
<Internet-source Link-name="Fire and Remote Sensing"
Url="../../library/firers.doc"/>
</Knowledge-object>
<Knowledge-object Type="ref-lib">
<Internet-source Link-name="Fire characteristics in Europe"
Url="../../library/fireeu.doc"/>
</Knowledge-object>
<Knowledge-object Type="ref-lib">
<Internet-source Link-name="Fire applications of MSG in Europe"
Url="../../library/fireapeu.doc"/>
</Knowledge-object>
```

Several more reference library objects. This list has been abbreviated. The objects in the reference library need not be HTML files. In fact it is common to use .doc, .pdf, .ppt, .jpg and .gif in the reference library.

```
</Environment>
```

End of the reference library environment.

```
<Environment Id="external-urls">
```

Start of the external URLs environment. This lists a number of external resources that will support the student in learning more about this topic.

```
<Knowledge-object Type="ext-url">
<Metadata>
<Title>Not used</Title>
</Metadata>
<Internet-source
Link-name="GOES Automated Biomass Burning Algorithm (ABBA)"
Url="http://cimss.ssec.wisc.edu/goes/burn/detection.html"/>
</Knowledge-object>
```

The definition of the first external URL. The Link-name is the title of the link as seen by the student and the Url is the full absolute path to the resource. The Metadata section may be omitted as it is not used.

```
<Knowledge-object Type="ext-url">
<Internet-source Link-name="Wildfire near San Diego, CA, USA"
Url="http://cimss.ssec.wisc.edu/goes/misc/20020210/020210.html"/>
</Knowledge-object>
```

```
<Knowledge-object Type="ext-url">
<Internet-source
Link-name="Fires in Virginia and North Carolina, USA"
Url="http://cimss.ssec.wisc.edu/goes/misc/970923.html"/>
</Knowledge-object>
```

Several more external URLs. This list has been truncated.

```
</Environment>
```

End of external URLs section.

```
<Environment Id="communication">
```

Start of communications environment. This environment is not supported and is included only for future expansion.

```
<Communication-object Id="mail">
<Send-mail>
<Send-to/>
</Send-mail>
</Communication-object>
<Communication-object Id="discussion">
<Asynchronous-conference>
<Participant>
<Role-ref/>
</Participant>
<Conference-manager>
<Role-ref/>
</Conference-manager>
</Asynchronous-conference>
</Communication-object>
<Communication-object Id="chat">
<Synchronous-conference>
<Medium>
<Text-conference/>
</Medium>
<Participant>
<Role-ref/>
</Participant>
<Conference-manager>
<Role-ref/>
</Conference-manager>
</Synchronous-conference>
</Communication-object>
</Environment>
```

End of communication environment.

```
<Activity Id="page1">
<Environment>
<Knowledge-object-ref Ref-worldwide-unique-id="intro.xml"/>
</Environment>
<Activity-description><!--Not used in MSG-CAL--><What/>
<Completed>
<User-choice/>
</Completed>
```

```
</Activity-description>
</Activity>
```

Definition of the first page/activity. The file name should be change to the name of the XML file which contains the definition of the knowledge object. This file should be placed in the unitf study subdirectory alongside this struct.xml file.

```
<Activity Id="page2">
<Environment>
<Knowledge-object-ref Ref-worldwide-unique-id="charact.xml"/>
</Environment>
<Activity-description><!--Not used in MSG-CAL--><What/>
<Completed>
<User-choice/>
</Completed>
</Activity-description>
</Activity>
<Activity Id="page3">
<Environment>
<Knowledge-object-ref Ref-worldwide-unique-id="locat1.xml"/>
</Environment>
<Activity-description><!--Not used in MSG-CAL--><What/>
<Completed>
<User-choice/>
</Completed>
</Activity-description>
</Activity>
<Activity Id="page4">
<Environment>
<Knowledge-object-ref Ref-worldwide-unique-id="locat2.xml"/>
</Environment>
<Activity-description><!--Not used in MSG-CAL--><What/>
<Completed>
<User-choice/>
</Completed>
</Activity-description>
</Activity>
<Activity Id="page5">
<Environment>
<Knowledge-object-ref Ref-worldwide-unique-id="concl.xml"/>
</Environment>
<Activity-description><!--Not used in MSG-CAL--><What/>
<Completed>
<User-choice/>
</Completed>
</Activity-description>
</Activity>
```

Include one activity definition for each page. For fewer pages remove the elements between <Activity Id=??> and </Acticity>. For more pages, although 5 is probably enough for one unit of study, duplicate the activity section for one page.

```
</Content>
```

End of content section containing all environments.

```
<Method>
```

```
<Activity-structure Id="module">
<Activity-sequence>
<Activity-ref Id-ref="page1"/>
<Activity-ref Id-ref="page2"/>
<Activity-ref Id-ref="page3"/>
<Activity-ref Id-ref="page4"/>
<Activity-ref Id-ref="page5"/>
</Activity-sequence>
</Activity-structure>
<Play>
<Role-ref Id-ref="learner"/>
<Activity-structure-ref Id-ref="module"/>
</Play>
</Method>
```

Method section defining the linear sequence of pages. The set of Activity-ref elements must match the number of pages defined in the Content section and the Ids in the Content section must match the Id-refs in this section. Only a linear structure is permitted in units of study so do not change the Activity-sequence to a Activity-selection

```
</Unit-of-study>
```

End of unit of study description.

Summing up, the unit-of-study template allows the author to define the services for this unit of study and the number and location of the pages contained in this unit of study. However, the actual pages are stored as separate knowledge objects. This means that individual pages could, in principle, be reused in different contexts.

Each page is created as a separate knowledge-object. A new page can be produced from a knowledge object template file. In order to minimise the effort in creating new interactions it is best to use an existing knowledge object XML file as a template. To obtain a template first go to the page. This might be a page in an exiusting unit-of-study or a page in the interactions showcase section. Right-click on the frame containing the page and use "Properties" or "View frame info" or simply select "Open frame in a new window". The object of this is to find the URL of this frame. Once you have found that URL (e.g. http://msgcal.eumetsat.de/msgcal/www/english/msg/interact/hotspot/hsex1.htm) you can cut and paste that URL into the location box of your browser but use the extension *.xml* in place of *.htm* and *sourc*e in place of *www*. This will cause the XML source file to be displayed in the browser. You can then save this page into your source directory (using "Save as…") to act as your new template. There is no restriction on what you call the file but it must have a .xml extension. Note that IE displays XML pages much better than Netscape but you can verify that Netscape has the full detail of the page by using "View page source" which displays the XML as well as IE.

The knowledge objects are based, as are the modules and units-of-study, on EML (Educational Modelling Language) but the interactions supported by EML are insufficient to support the rich level of interactivity required in MeteoCAL. As a result, the interactions from the IMS QTI[5] have been added to enable a wider range of question types. The IMS QTI component is identified in EML files as *questtestinterop* in the *interactions* section. Full details of how to edit every template are not provided here as it would take too much space and each template is rather self-explanatory. However, as an introduction the following is an example of editing a knowledge object that holds a multiple choice question.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Do not modify this first line.

[5] QTI: Question and Test Interoperability (see http://www.imsproject.org/question/index.html)

```
<Knowledge-object>
<Metadata>
<Title>Deriving the final product</Title>
</Metadata>
```

Change the title to describe your knowledge object.

```
<Source>
<Section>
<Source>
```

Start of introductory section.

```
<P>We will use the example of the  Atlantic Sea Surface Temperature
(ASST) product.  MSG data is used to calculate the skin temperature of
the sea surface.  The ASST is calculated using a simple skin
temperature to bulk temperature transformation.</P>
<P>Information from the sea surface passes through a cloudless
atmosphere into space in only certain well defined bands of the
spectrum.  Three SEVIRI channels IR 3.9, IR 8.7,  IR 10.8  are in such
transparent  bands.
</P>
```

Introductory paragraphs. Several others have been removed to make this example shorter.

```
<Figure>
<Figure-source Ref-worldwide-unique-
id="../../../../resource/msg/calib/sst_example/safoqklnarsst_20020531_a
sc_withscale.gif"/>
<Figure-text>
<P>GOES SST over the East Atlantic, 31st May 2002 </P>
</Figure-text>
</Figure>
```

Figure definition including caption.

```
</Source>
</Section>
<Section>
```

End of introductory section.

```
<Source>
<Interactions>
```

Start of interaction section.

```
<questestinterop>
```

Start of a question.

```
<item ident="question1" label="mcqmr">
```

New question. Unique identifier question1 will distinguish this question from others. Though this example contains only one question. To add additional questions copy everything between <item> and </item> and

identify it as question2, quiestion3 etc. The label mcqmr identifies this question as a multiple choice question with multiple responses.

```
<presentation>
```

Start of the presentation of the question.

```
<material>
<mattext label="question">Which of the following operations have been performed to create the above image ?
</mattext>
</material>
```

Presents the general text of the question. Use <matimage> if an image is to be included as part of the question. See Appendix B for more details.

```
<response_lid ident="mcqmr" rcardinality="Multiple">
```

Begin defining response labels with multiple correct answers.

```
<render_choice>
```

This question will be rendered in the style of a multiple choice question.

```
<response_label labelrefid="correct" ident="a">
<material>
<mattext>
Land mask
</mattext>
</material>
</response_label>
```

The first selectable response. The labelrefid is used to identify this as the correct answer. The ident “a” is used to provide a unique identifier for this selected response so that later appropriate feedback can be linked to the selection. Only the part inside the <mattext> element, in this case “Land mask” is shown on screen.

```
<response_label  labelrefid="incorrect" ident="b">
<material>
<mattext>
Face mask
</mattext>
</material>
</response_label>
```

Another selectable response. In this case an incorrect answer.

```
<response_label  labelrefid="incorrect" ident="c">
<material>
<mattext>
Wind mask
</mattext>
</material>
</response_label>
```

Another incorrect answer.

```
<response_label  labelrefid="correct" ident="d">
<material>
<mattext>
Cloud mask
</mattext>
</material>
</response_label>
```

A fourth response. In this case the answer is also correct.

```
</render_choice>
</response_lid>
</presentation>
```

End of the section in which the choices are rendered and that completes the presentation of the question on screen.

```
<itemfeedback ident="a">
<material>
<mattext>Yes, all data over a land mass has been removed.
</mattext>
</material>
</itemfeedback>
```

Feedback provided when response ident=”a” is selected.

```
<itemfeedback ident="b">
<material>
<mattext>No !
</mattext>
</material>
</itemfeedback>
```

Feedback provided when response ident=”b” is selected.

```
<itemfeedback ident="c">
<material>
<mattext>No.
</mattext>
</material>
</itemfeedback>
```

Feedback provided when response ident=”c” is selected.

```
<itemfeedback ident="d">
<material>
<mattext>Yes, all data where the sea is obscured by cloud has been
removed.
</mattext>
</material>
</itemfeedback>
```

Feedback provided when response ident=”d” is selected.

```
</item>
```

```
End of the first (and in this case only) question.

</questestinterop>
</Interactions>
```

End of question interaction.

```
</Source>
</Section>
</Source>
</Knowledge-object>
```

End of this knowledge object

```
<!-- Keep this comment at the end of the file
Local variables:
mode: xml
sgml-omittag:nil
sgml-shorttag:nil
sgml-namecase-general:nil
sgml-general-insert-case:lower
sgml-minimize-attributes:nil
sgml-always-quote-attributes:t
sgml-indent-step:2
sgml-indent-data:t
sgml-parent-document:nil
sgml-exposed-tags:nil
sgml-local-catalogs:nil
sgml-local-ecat-files:nil
End:
-->
```

Keep this part at the end of the file unchanged.

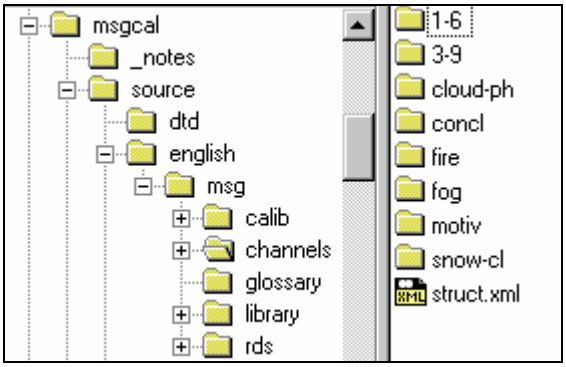
Once you have created a single knowledge object it can be transformed to HTML. On the other hand you can build all the knowledge objects you want to include in a single unit-of-study and transform them all at once if you prefer. In fact you can even transform several modules units of study or even several modules all at once.

In order to process the knowledge object or the unit-of-study, see the section on Transforming Source Files.

5: Modify the structure of an existing module

To modify an existing module by adding new units-of-study or replacing existing units of study with new versions it is simply a matter of modifying the struct.xml file at the module directory level. Let's assume for example that we want to remove the fire detection unit of study from the channels module and replace it with one on determining night-time temperatures.

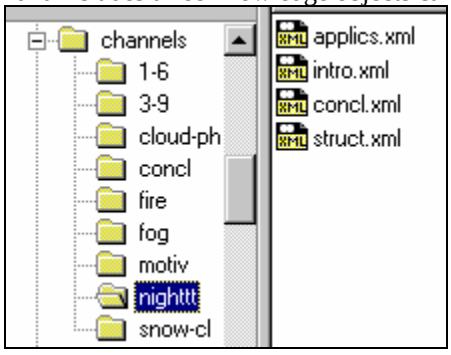
The directory structure of the channels modules is shown below:



and the file struct.xml contains the structure of the module. The relevant part of that file is the <Activity-structure>

```
<Activity-structure Default-visibility="Show" Id="module">
<Activity-sequence Link-name="module">
<Activity-sequence>
<Unit-of-study-ref Link-name="Motivation" Ref-worldwide-unique-id="motiv/struct.xml"/>
</Activity-sequence>
<Activity-selection Number-to-select="2">
<Unit-of-study-ref Link-name="1.6 micrometre channel" Ref-worldwide-unique-id="1-6/struct.xml"/>
<Unit-of-study-ref Link-name="3.9 micrometre channel" Ref-worldwide-unique-id="3-9/struct.xml"/>
</Activity-selection>
<Activity-selection Number-to-select="4">
<Unit-of-study-ref Link-name="Snow-cloud discrimination" Ref-worldwide-unique-id="snow-cl/struct.xml"/>
<Unit-of-study-ref Link-name="Cloud phase determination" Ref-worldwide-unique-id="cloud-ph/struct.xml"/>
<Unit-of-study-ref Link-name="Fog detection" Ref-worldwide-unique-id="fog/struct.xml"/>
<Unit-of-study-ref Link-name="Fire detection" Ref-worldwide-unique-id="fire/struct.xml"/>
</Activity-selection>
<Activity-sequence>
<Unit-of-study-ref Link-name="Conclusions" Ref-worldwide-unique-id="concl/struct.xml"/>
</Activity-sequence>
</Activity-sequence>
</Activity-structure>
```

and the line containing the reference to the fire unit-of-study is highlighted. An editor, either an XML editor or a normal text editor can be used to change this line. The new unit-of-study (nighttt) has been added to the directory structure and it includes three knowledge objects called intro, applics and concl as



well as a struct file which defines the sequence of these three as well as the reference library, communications and external URL resources for this new unit-of-study. To change the structure of the module the line

```
<Unit-of-study-ref Link-name="Fire detection" Ref-worldwide-unique-id="fire/struct.xml"/>
```

now needs to be changed to

```
<Unit-of-study-ref Link-name="Night-time temperatures" Ref-worldwide-unique-id="nighttt/struct.xml"/>
```

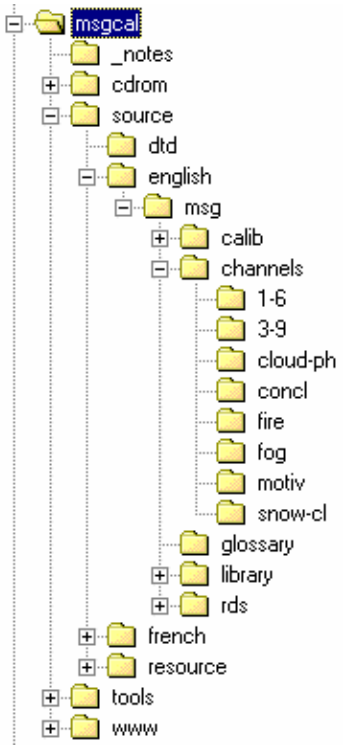
The new module can now be transformed (see the section Transforming Source Files) and the www and cdrom versions will be produced.

As an interesting exercise, see if you can work out how you would add the night-time temperatures unit-of-study without removing the fire detection unit-of-study, so that the module has a choice of five different applications of these new MSG channels. The solution is given below.

```
<Activity-structure Default-visibility="Show" Id="module">
<Activity-sequence Link-name="module">
<Activity-sequence>
<Unit-of-study-ref Link-name="Motivation" Ref-worldwide-unique-id="motiv/struct.xml"/>
</Activity-sequence>
<Activity-selection Number-to-select="2">
<Unit-of-study-ref Link-name="1.6 micrometre channel" Ref-worldwide-unique-id="1-6/struct.xml"/>
<Unit-of-study-ref Link-name="3.9 micrometre channel" Ref-worldwide-unique-id="3-9/struct.xml"/>
</Activity-selection>
<Activity-selection Number-to-select="4">
<Unit-of-study-ref Link-name="Snow-cloud discrimination" Ref-worldwide-unique-id="snow-cl/struct.xml"/>
<Unit-of-study-ref Link-name="Cloud phase determination" Ref-worldwide-unique-id="cloud-ph/struct.xml"/>
<Unit-of-study-ref Link-name="Fog detection" Ref-worldwide-unique-id="fog/struct.xml"/>
<Unit-of-study-ref Link-name="Fire detection" Ref-worldwide-unique-id="fire/struct.xml"/>
<Unit-of-study-ref Link-name="Night-time temperatures" Ref-worldwide-unique-id="nighttt/struct.xml"/>
</Activity-selection>
<Activity-sequence>
<Unit-of-study-ref Link-name="Conclusions" Ref-worldwide-unique-id="concl/struct.xml"/>
</Activity-sequence>
</Activity-sequence>
</Activity-structure>
```

6: Server-side architecture

The MSG-CAL files and directory structure are provided on CD. The directory structure at this location holds the **source** files (i.e. the files containing the XML content of the modules), the **www** files (i.e. the files containing HTML which are the visible part of the MSG-CAL web site and the **tools** required to convert source files to HTML files. Another directory at the same level as **www** contains the **cdrom** files (i.e. the files making up the CD-ROM specific version of the MSG-CAL modules). The resources for all modules and units of study are in the **resource** directory which has the same subdirectory structure as **english** and **french**. The modules are **calib**, **channels** and **rds**. Each of these modules contains its units-of-study as subdirectories. Each unit-of-study subdirectory contains the knowledge object files. All reference library files for all units of study are stored in the **library** directory. The directory structure is shown below:



An installation of MSG-CAL or new MeteoCAL modules should have this directory structure. The source files (all XML files) will be transformed into HTML files in both the www and cdrom directories by one of two processes. This can be achieved either through the msgmaintain tool (a Tomcat application available only on the EUMETSAT web site) or by intraMediate (a command line processor which can be run from a batch file or shell script).

It is assumed that this directory structure exists. The location of the msgcal directory is referred to as the project root. The main tool for transformation, intramediate, is in the tools directory which contains a subdirectory called resources. In this directory an XML called intramediate.xml contains the following information:

```
<intramediate>
<projectRoot>///barbar/apache/htdocs/devel/msgcal/</projectRoot>
<debug>false</debug>
</intramediate>
```

The <projectRoot> should be modified to define the exact path on the local machine at which the MSGCAL: directory structure is located. If <debug> is set to true the considerable amounts of debugging information will be produced. It is worth setting it to true initially.

The bin directory contains the script that can be used to call the intramediate tool. Two versions are provided intramediate.sh for Unix and intramediate .bat for PCs.

PC version
The batch file contains:

```
@ echo off

set INTRAMEDIATE_HOME=%~dp0..

set INTRAMEDIATE_LIB=%INTRAMEDIATE_HOME%\lib
set INTRAMEDIATE_RESOURCES=%INTRAMEDIATE_HOME%\resources

set CLASSPATH=

set CLASSPATH=%CLASSPATH%;%INTRAMEDIATE_RESOURCES%
set CLASSPATH=%CLASSPATH%;%INTRAMEDIATE_LIB%\xalan.jar
set CLASSPATH=%CLASSPATH%;%INTRAMEDIATE_LIB%\intramediate.jar
set CLASSPATH=%CLASSPATH%;%INTRAMEDIATE_LIB%\jdom.jar
set CLASSPATH=%CLASSPATH%;%INTRAMEDIATE_LIB%\xml-apis.jar
set CLASSPATH=%CLASSPATH%;%INTRAMEDIATE_LIB%\xercesImpl.jar

rem echo using classpath:%CLASSPATH%

set CMD_LINE_ARGS=%*

rem echo command line args:%CMD_LINE_ARGS%

java -classpath %CLASSPATH% com.intrallect.intramediate.Intramediate
www %CMD_LINE_ARGS%
```

IntraMediate is a Java application and so Java must be available on the machine on which it is run. In most cases this batch file need not be modified. It works on Windows NT and 2000 but on some other versions of Windows it may be necessary to modify the line
`set CMD_LINE_ARGS=%*`
to
`set CMD_LINE_ARGS=%1`
in which case a single file or directory can be processed but not a list of files. The www in the java command specifies that the transformation will be from the source into the www version. It could also be cdrom or www:cdrom for both.

The entire intramediate directory can be moved to another machine as long as the intramediate.xml file points to the location of the project root.

A shortcut to the batch file can be created and [placed on the users desktop for maximum convenience. Any file or directory dropped onto the shortcut will be processed as the name of the file or directory will be passed as a parameter.

Unix version
The script file contains:

```
#!/bin/sh
```

```
# ------------------------------------------------------------------
# runner for intramediate
#
#           Environment Variable Prequisites
#
#     JAVA_HOME Must point at your Java Development Kit installation.
#
# ----- Execute The Requested Command ------------------------------
#
#           ARG $1 is transformation types seperated by colon
#           ARG $2..$3.. etc are paths or files
#            ( or combination of both ) to be transformed
#

#INTRAMEDIATE_HOME="$0"/../..
INTRAMEDIATE_HOME=/devel/xml/msgcal/tools/intramediate
CLASSPATH="$INTRAMEDIATE_HOME"/resources:"$INTRAMEDIATE_HOME"/lib/xalan
.jar:"$INTRAMEDIATE_HOME"/lib/intramediate.jar:"$INTRAMEDIATE_HOME"/lib
/jdom.jar:"$INTRAMEDIATE_HOME"/lib/xml-
apis.jar:"$INTRAMEDIATE_HOME"/lib/xercesImpl.jar

echo "Using CLASSPATH: $CLASSPATH"
#echo "Using INTRAMEDIATE_HOME: $INTRAMEDIATE_HOME"

exec java -classpath "$CLASSPATH"
com.intrallect.intramediate.Intramediate $1 $2 $3 $4 $5 $6 $7 $8 $9
```

It is assumed that a JAVA_HOME environment variable exists which points to the Java Development Kit location. INTRAMEDIATE_HOME must be modified to point to the intramediate directory.

The script is used by calling the script name with parameters:

```
intramediate.sh www channels
```

to transform all files in the channels module to www. Or it might be convenient to alias the "intramediate.sh www" part to a simple command such as www_transfrom which then only needs a file or directory or list of files and directories.

7: Transforming Source Files

Once you have created knowledge objects, units-of-study or modules in the appropriate directory structure in the source part of the msgcal directories, you will want to transform them into HTML files in the www and/or cdrom directories.

On a PC this is as simple as dragging the file or directory onto the batch file (or shortcut to the batch file) which the administrator has set up.

On Unix you need to use the command defined by the administrator followed by a list of files or directories to be transformed.

It is as simple as that! The resulting files will appear in the www directory or one or more error message will appear.

8: Making Modules/Courses Visible on a Web Server

In order to make a new module or a new course visible on a web server you need to install the www directory on a web server, or ensure that the location of the www directory is part of your web server file space.

Then you need to create an entry point to the module/course. This entry point needs to carefully set up so that all the features of the MeteoCAL environment are made available, for example the reference library and glossary and the ability to switch languages. (Note that if you do not want any of these features it is possible to link directly to the individual knowledge object HTML files. However, it is not possible to use the unit-of-study index.html file without the full MeteoCAL environment to handle the navigation.) See Appendix F for setting the languages your course will be delivered in; the default is English and French.

The place where the environment is set up is in the msgcal/www/english/courses directory (or the equivalent in French or any other language). To set up a new course, make a copy of the directory msgcrs in the courses directory. Rename this copied directory to something appropriate for your particular course. Then, to link modules into this course edit the crsindex.htm file (found inside your new course directory) as follows:

crsindex.htm:
Edit the file so that the entries near the end of the file are linked to the index.htm files for each of your modules (or only one if you have only one module). For example, this section is taken from the crsindex.htm file for the MSG-CAL modules so contains three references to modules:

```
<h4><a href="../../msg/rds/index.htm" target="content"><img
src="../../../icons/module.gif" width="43" height="11" border="0">
Rapidly Developing Storms</a></h4>
<h4><a href="../../msg/channels/index.htm"><img src="../../../icons/module.gif"
width="43" height="11" border="0"> 1.6 and 3.9 micrometer channels</a></h4>
<h4><a href="../../msg/calib/index.htm"><img src="../../../icons/module.gif" width="43"
height="11" border="0"> Calibration</a></h4>
```

Change the file so that it points to the directories continuing your module(s), for example if there were three modules in the course called module1, module2, and module3 the file should now read:

```
<h4><a href="../../msg/module1/index.htm" target="content"><img
src="../../../icons/module.gif" width="43" height="11" border="0">
Module1 title</a></h4>
<h4><a href="../../msg/module2/index.htm"><img src="../../../icons/module.gif" width="43"
height="11" border="0"> Module2 title</a></h4>
<h4><a href="../../msg/module3/index.htm"><img src="../../../icons/module.gif" width="43"
height="11" border="0"> Module3 title</a></h4>
```

If you have only one module then you will have only one entry, not three.

In this file you may also wish to edit the text that will appear on the page such as the following from the crsindex.htm file (you can edit any of the parts shown here in italics:

```
<h1>Meteosat Second Generation (MSG) </h1>
<p><b> These modules should be viewed in Microsoft Internet Explorer 5.5
and above or Netscape 6.2 and above.</b></p>
<p> Welcome to the the MSG training modules. To use the modules, click
on one of the links to the right of this text. If this is your first
time using these materials, you may want to look at the help which
you can get to by clicking the top icon to the left of this text.
The help describes what the other icons on the left do and how to
use some of the interactive elements of the modules.</p>
```

Linking to a web site

To complete the process of linking to a web site you can create a web page anywhere (even on another web site) which includes a link to your mycourseindex.htm file, for example

```
<p> Please try this new course.</p>
<p><a href="msgcal/www/english/courses/msgcrs/mycourseindex.htm"> My Course </a></p>
```

Note that the actual path in the HREF attribute will depend on the location on your web server and where the file containing this sits on your web server.

9: Choosing and Using an XML Editor

This section describes two options. The first is using a strict XML editor which checks everything you type against the DTD. This means that it checks both the well-formedness and validity of the XML file as you type. The second is a programmer's editor which helps you spot errors in the well-formedness of an XML file by highlighting sections. It also allows you to process files without leaving the editor.

Strict XML Editor

It is sensible to use an XML editor to produce XML files for several reasons:

1. XML files have a complex structure and must be "well-formed". It is easy to make typing mistakes that can have far-reaching effects.
2. XML files for MSG-CAL must be "valid". That is, they must match the structure laid out in the appropriate DTD (document type definition) file. It is easy to make typing mistakes that can have far-reaching effects.

You can choose any XML editor. Most XML editors will ensure that the XML is well-formed. However, not all editors will provide validation against a DTD and report errors. Two editors which have been used and found to be suitable are:

1. XML Spy: XML Spy is a commercial product
2. Xeena: Xeena is free for personal use and is available from IBM AlphaWorks (http://www.alphaworks.ibm.com/tech/xeena)

The following assumes that Xeena is used, partly because it is free but also because it is more intuitive to use.

Setting up Xeena: The information that comes with Xeena is not very helpful but it comes complete with several batch files (bml.bat, configdtd.bat, etc). Take one of these and edit it to produced the following and called it eml.bat:

```
@echo off
set XEENA_HOME=D:\IBM\Xeena
rem Run XEENA to edit any XML Schema (version Dec 1999)
rem
rem Brought to you by the IBM Haifa Research Lab, Israel
rem Please send feedback to helpmap@il.ibm.com
rem (C) 1999-2000 IBM Corporation. All rights reserved.
rem

%XEENA_HOME%\xeena -dtd "D:\intrallect\msg-cal\eml\eml10.dtd" -root "Unit-of-study"
```

Only the last line needs to be edited. The first part in quotes needs to refer to the DTD you want to use and the second part needs to refer to the root term in that DTD.

Dire warning: If you copy XML files to use them in Xeena make sure you edit the second line to include the location of your version of the DTD or Xeena will grab 100% of your CPU until you kill it.

Programmer's Editor

There are many different varieties of programmer's editor but one that is particularly convenient for MeteoCAL use is UniRed[6] an open source, Unicode editor for all Windows platforms. This editor is capable of recognising XML files and will highlight opening and closing element tags. It also includes "compiler" commands and these can be adapted to run the MeteoCAL processor (intramediate) from the editor menu. This editor also supports a spelling checker (iSpell) with multiple language versions.

[6] http://www.esperanto.mv.ru/UniRed/ENG/

Install the editor simply download and install in the usual way. To run the processor from the editor menu edit the xml.ini file in the Options menu, Compile/*.ini item to read:

```
[intramediate]
cmd=java -classpath
D:\msgcal\tools\intramediate\lib\intramediate.jar;D:\msgcal\tools\intram
ediate\lib\xalan.jar;D:\msgcal\tools\intramediate\lib\jdom.jar;D:\msgcal
\tools\intramediate\lib\xml-
apis.jar;D:\msgcal\tools\intramediate\lib\xercesImpl.jar;D:\msgcal\tools
\intramediate\resources com.intrallect.intramediate.Intramediate www
%file%
show=1
home=D:\msgcal\tools\intramediate\bin
```

Note that:

1. The section starting cmd= and ending %file% must all appear on a single line without line breaks.
2. The path D:\msgcal\tools\intramediate should be changed to show where you have installed the tools on your computer, see appendix C.

Once this file is installed as your xml.ini and the editor has been re-started you should find that when you are editing a MeteoCAL source (XML) file you will be able to use the external command item on the run menu, labelled “intramediate” to process the file. In this way you can edit the file in UniRed and process it while at the same time having a browser open to view the results.

Appendix A: Creating storyboards

Storyboards are a useful way of capturing your ideas for a module without having to think about the technology. Below are three templates for storyboards, one each for module, unit of study and knowledge-object. Since one module might have 6 units-of-study and each unit-of-study might have 3 knowledge objects you could use the module storyboard once, the unit of study storyboard 6 times and the knowledge object storyboard 18 times.

These storyboards can be cut and pasted into word processor documents so that they can be easily edited.

Module storyboard template (replace all sections in italics with your own text)

Title: *short text here*

Learning objective: *text description of a learning objective*
Learning objective type: select one of attitude/ competence/ insight/ knowledge/ skill/ other
– *repeat for each learning objective.*

Prerequisite: *text description of required prior experience/knowledge/skill*
Prerequisite type: select one of attitude/ competence/ insight/ knowledge/ skill/ situational;-factor/ other
– *repeat for each prerequisite.*

Module description*: text that describes the way a student might use this module. This is a good place to refer to the roles of the different units-of study as this text will be displayed each time the student has to choose the next unit-of study.*

Units of study: - *repeat for each unit of study*
Title*: short text title for the unit of study*
Description: *text that will appear to describe this unit of study to the student when he/she is given the option of choosing the next unit of study. This is text that specifically relates to the way this unit of study can be used in this module and this is the only place where that contextual information can be provided.*
Subdirectory: the *name of the subdirectory in which the unit-of-study will be placed.*
Structure: *Sequence/ Selection- if selection which other units of study are in the selection. A selection should be chosen when you want to offer more than one unit of study and there is complete freedom for the student to choose one route or another.*

Unit of study storyboard template (replace all sections in italics with your own text) – repeat for each unit-of-study

Title: *short text here*

Learning objective: *text description of a learning objective*
Learning objective type: select one of attitude/ competence/ insight/ knowledge/ skill/ other
– *repeat for each learning objective.*

Prerequisite: *text description of required prior experience/knowledge/skill*
Prerequisite type: select one of attitude/ competence/ insight/ knowledge/ skill/ situational;-factor/ other
– *repeat for each prerequisite.*

Reference library: extend table for number of objects in library

Title	Document file name
Name the student should see	*.doc, .ppt, .pdf, .html, .gif, .jpg or any other file that will support the learner and provide additional information*

External URLs: extend table for number of URLs

Title	Absolute URL
Name the student should see	*Full URL of the resource*

Knowledge objects to be used in this unit of study:
List all file names used to store knowledge objects in XML.

Knowledge object storyboard template (replace all sections in italics with your own text)

- repeat for each knowledge object

It is difficult to create a general storyboard for all types of knowledge object. The general structure of most knowledge objects is for an introductory section to be followed by an interaction section.

Introduction section:
Write the text that should appear on screen.
If an figure is to be included note the *name of the file* containing the image and the *caption* to be used

Interaction section:
Select the interaction (only one) and complete the relevant sections:

Multiple choice – one or more questions. Copy and add the shaded area for each new question.

Multiple choice question			
Repeat shaded section for the number of questions in the interaction	Text of question	Image file name if included	Image caption if included
Repeat this row for the number of responses to the question	Text of response on screen	Feedback for this response	Correct/incorrect

Matrix questions – up to five columns (or add more) with as many options as you wish (add new rows iof more than five). For each new section add another copy of the shaded area.

Matrix question					
Repeat shaded section for the number of questions in the interaction	Text of question	Image file name if included	Image caption if included		
Response labels	Text of first column	Text of second column	Text of third column	Text of fourth column	Text of fifth column
Option1 text	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect
Option 2 text	Feedback for this selection	Feedback for this selection	Feedback for this selection	Feedback for this selection	Feedback for this selection

	Correct/incorrect	Correct/incorrect	Correct/incorrect	Correct/incorrect	Correct/incorrect
Option 3 text	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect
Option 4 text	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect
Option 5 text	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect	Feedback for this selection Correct/incorrect

Image hotspot: One or more images. If more than five images add columns. Copy hotspot rows for the number of hotspots of each type (circle, rectangle or polygon)

Image hotspot					
Text of question					
Image file names	Image 1	Image 2	Image 3	Image 4	Image 5
Image captions	Caption 1	Caption 2	Caption 3	Caption 4	Caption 5
Hotspot (repeat for other hotspots) circle/ellipse	Correct/incorrect	Feedback	Centre x	Centre y	radius
Hotspot (repeat for other hotposts) rectangle	Correct/incorrect	Feedback	Top left x,y	Bottom right x,y	
Hotspot (repeat for other hotposts) polygon/bounded	Correct/incorrect	Feedback	X1,Y1	X2,Y2	X3,Y3 etc

Animation					
Title text	File type	directory	Start image	Stop image	

Slide show		
Repeat for each slide	Image file name	Image caption

Text drag and drop: Add more columns if more than five draggables and more rows if more than five target areas.

Text drag and drop					
Text of question	Image file name if included	Image caption if included	width	height	Offset x,y
	Draggable 1	Draggable 2	Draggable 3	Draggable 4	Draggable 5
	Intial x,y for 1	Intial x,y for 2	Intial x,y for 3	Intial x,y for 4	Intial x,y for 5
	Text	Text	Text	Text	Text
Target 1, x.y	Feedback for draggable 1 in target 1	Feedback	Feedback	Feedback	Feedback
Target 2 x.y	Feedback	Feedback	Feedback	Feedback	Feedback
Target 3 x.y	Feedback	Feedback	Feedback	Feedback	Feedback
Target 4 x.y	Feedback	Feedback	Feedback	Feedback	Feedback
Target 5 x.y	Feedback	Feedback	Feedback	Feedback	Feedback

Image drag and drop: Add more columns if more than five draggables and more rows if more than five target areas.

Imagedrag and drop					
Text of question	Image file name if included	Image caption if included	width	height	Offset x,y
	Draggable 1	Draggable 2	Draggable 3	Draggable 4	Draggable 5
	Intial x,y for 1	Intial x,y for 2	Intial x,y for 3	Intial x,y for 4	Intial x,y for 5
	Image file name	Image file name	Image file name	Image file name	Image file name
Target 1, x.y	Feedback for draggable 1 in target 1	Feedback	Feedback	Feedback	Feedback
Target 2 x.y	Feedback	Feedback	Feedback	Feedback	Feedback
Target 3 x.y	Feedback	Feedback	Feedback	Feedback	Feedback
Target 4 x.y	Feedback	Feedback	Feedback	Feedback	Feedback
Target 5 x.y	Feedback	Feedback	Feedback	Feedback	Feedback

Appendix B: MeteoCAL XML

This appendix describes in detail the XML (eXtensible Markup Language) on which MeteoCAL is based. The XML DTD (document type definition) used in MeteoCAL is based partly on the Open University of the Netherland's EML (Educational Modelling Language), partly on IMS Global Learning Consortium's QTI (Question and Test Interoperability) and partly on interactions developed specifically for meteorological purposes by Intrallect. This combination could be used in a huge variety of different ways. The purpose of this appendix is to define the ways in which the MeteoCAL tools expect the XML files to be constructed.

The DTDs are a set of files which can be found in the source/dtd section of the directory structure. The DTDs themselves are not discussed here.

The first part of the appendix is a diagram showing the relationship of each XML element. Following that is a detailed description of all the supported elements and their attributes and purpose.

Elements
Elements in XML have the form:

```
<Knowledge-object>
… contents of the Knowledge object – usually text and other elements
</Knowledge-object>
```

or <Situational-factor/> if the element has no contents.

Attributes
Attributes are set inside the opening element term as in:

```
<response_label ident="N" labelrefid="incorrect"
rarea="Bounded">7,479,47,487,45,528,7,527</response_label>
```

where 7,479,47,487,45,528,7,527 is the content of the response_label element and ident="N" labelrefid="incorrect" rarea="Bounded" are its attributes.

Table 4: Complete list of all supported XML elements in a Unit-of-Study, showing their hierarchical relationship.

Unit-of-study
Metadata
Title
Roles
Learning-objectives
Learning-objective *Id*
Objective-description
P
Objective-type
Attitude
Competence
Insight
Knowledge
Other
Skill
Prerequisites
Pre-requisite
Prerequisite-description
P
Prerequisite-type
Attitude
Competence
Insight
Knowledge
Other
Skill
Situational-factor
Content
Activity
Environment *Id=ref-library*

	Knowledge-object	*Type=ref-lib*		
	Internet-source	*Link-name*	*Url*	
Environment	*Id=external-urls*			
	Knowledge-object	*Type=ext-url*		
	Internet-source	*Link-name*	*Url*	
Environment				
	Knowledge-object-ref	*Ref-worldwide-unique-id*		
Environment				
	Knowledge-object			
	Metadata			
	Title			
	Source			
	P			
Method				
Activity-structure				
Activity-sequence				
	Unit-of-study-ref	*Link-name*	*Ref-worldwide-unique-id*	*popup*
Activity-selection	*Number-to-select*			
	Unit-of-study-ref…			
Play				
Role-ref	*Id-ref*			
Activity-structure-ref	*Id-ref*			

Table 5: Complete list of all supported XML elements in Knowledge Object, showing their hierarchical relationship.

| | | |

			render_hotspot				
				response_label[7]	*ident*	*labelrefid=correct|incorrect*	*match_group*
				response_label[8]	*ident*	*labelrefid=correct|incorrect*	*rarea=Ellipse|Rectangle|Bounded*
				material…			
			response_grp	*ident*			
				render_extension			
	List	*Ordered=yes|no*					
		Li					
	P						
	Special	*Content-type*	*Link-name*	*Position*			

[7] as used in Drag and Drop exercises
[8] as used in Hotspot exercises

Table 6: XML elements used in MeteoCAL. Note that the use of upper and lower case in element names is important.

Element	Parent element	Description	Attributes[9]
Activity	Content		
Activity-selection	Activity-structure	Container for one or more <Unit-of-study-ref> which will be taken in sequence. Used in the module struct.xml file only.	*Number-to-select = "number of items in the selection"*
Activity-sequence	Activity-structure	Container for one or more <Unit-of-study-ref> which will be offered as choices at the same stage. Used in the module struct.xml file only.	
Activity-structure	Method	Container within which <Activity-sequence> and <Activity selection> are used to defined navigation through units-of-study in the module struct.xml files. Complex navigation can be built using nesting of <Activity-sequence> and <Activity selection>	
Activity-structure-ref	Play	Used for completeness with EML	Id-ref = "module" Required
Animation-title	Image-animation or Image-step-frame	Contains text which will form the title for the animation. Can be empty, i.e. <Animation-title/>	
Asynchronous-conference	Communication-object	Used only to offer a "hook" which could be used in the future.	
Attitude	Objective-type or Prerequisite-type	Used as an empty element.	
Communication-object	Environment	Used in unit-of-study struct files to define the communications support. Not implemented in MSG-CAL but included for future expansion.	Id = discussion|mail|chat Required
Competence	Objective-type or Prerequisite-type	Used as an empty element.	
Content	Unit-of-study	Used as a container for <Environment> in struct.xml files	
Environment	Activity	1. Used in module struct files to contain the single <Knowledge-object> which acts to introduce the module	Id = ref-library|external-urls|communications Omitted when containing a <Knowledge-object> or

[9] Attributes separated by | (e.g ref-lib|ext-url) are alternatives, one of which should be used. Attributes in italics (e.g. "*relative path of .xml file containing knowledge-object source*") describe the text that should be used. Note that all attributes should be surrounded by quotes.

		and describes the relationship of the various units-of-study. 2. Used in unit-of-study struct files to define the various types of environment. These are either one of the three standard types given by the Id attribute or, within the <Activity> associated with each page in the Unit-of-study, to act as a container for the <Knowledge-object-ref>.	<Knowledge-object-ref>, otherwise required.
Figure	Source	Used in Knowledge-object files to include figures. Not used within questions where <material> should be used.	Id = "*an identifier unique in this file*" Required.
Figure-source	Figure	Used to define the file which contains the Figure.	Ref-worldwide-unique-id = "*relative path of image file in resource directory*" Required
Figure-text	Figure	Used as a caption for a <Figure> by including the caption in a <P> element.	None
Image-animation	Interactions	Used in a Knowledge object file to define an animation sequence. Requires an <Image-path> with <Animation-title> being optional.	width = "*width in pixels*" height = "*height in pixels*" start-image = "*number of the first image in the sequence defined by <Image-path>*" start-image = "*number of the first image in the sequence defined by <Image-path>"* Required
image-frame	Image-step-frame	Used to define each frame in an <Image-step-frame> or "slide show".	image-caption = "*text to appear as caption for the image in this frame*" Optional Image-path = "*relative path to image for this frame*" Required Launch-ref = ???
Image-path	Image-animation	Used to define where to find all the images for the animation. The file name should be of the form: path/file*nn*.*ext* where *nn* is the Image number and *ext* is image-type so the pathname only needs path/file.	Image-type = gif|png|jpg Path-name = "*relative path to images, excluding image number and file extension*"
Image-step-	Interactions	Used to form a "slide show"	width= width of images

frame		containing several <image-frame> elements.	height = height of images Required
Insight	Objective-type or Prerequisite-type	Used as an empty element.	
Interactions	Source	A container for an interaction. The choice of interactions is either questestinterop for interactions that pose questions to be answered, or Image-step-frame or Image-animation when the emphasis is on presenting image-based information.	
Internet-source	Knowledge-object	Used in unit-of-study struct.xml files to define the actual source or reference library and external URL material.	Link-name= “text used as a title for the reference item” Url=”location of the item, relative path for reference library and absolute url for external urls” Required
Item	questestinterop	Used to define questions. When several questions are included in an interaction there must be one <item> for each question. Each <item> must have a unique ident.	ident = “identifier unique within the file” Required label = ???? Optional
itemfeedback	item	A container for the <material> which will comprise the feedback to the appropriate <resonse_label>.	ident = “identifier identical to the <response_label> identifier to which the feedback corresponds” Required For mcq’s, matrix mcq’s and drag and drop: you can set generic feedback to appear to save you creating feedback for every possible option. To do this, create two itemfeedback tags, one with the ident=”correctfeedback” and the other ident=”incorrectfeedback”. For mcq’s and matrix mcq’s, the generic feedback should be set for each separate question.
Knowledge	Objective-type or Prerequisite-type	Used as an empty element.	
Knowledge-object	Environment or as the root element, it has no parent in knowledge object files.	1. Root element of a Knowledge-object file. 2. Used within the appropriate <Environment> of a unit-of-study struct file	Type = ref-lib|ext-url Omitted when the root element of a Knowledge-object file, otherwise required.

		to contain the <Internet-source> for reference library and external urls.	
Knowledge-object-ref	Environment	Used in unit-of-study struct files to define the names of knowledge object comprising each page.	Ref-worldwide-unique-id = "*relative path of .xml file containing knowledge-object source*" Required
Learning-objective	Learning-objectives	Container for a single learning objective.	
Learning-objectives	Unit-of-study	Container for a set of learning objectives.	
Li	List	Contains an element of a list.	
List	Source	Used to create either ordered or unordered lists.	ordered=yes|no
material	presentation or response_label or itemfeedback	Used as a general container in different parts of a question.	label="|bgimage|congrats" Optional unless used to specify that the material contains the background image or congratulations message (on completion) for a drag and drop question.
matimage	material	Used to show an image as part of <material> either in a question or in one of the responses to a question or in the feedback to a question.	label = "*if set to same label as mattext links image and caption*" Optional width= "image-width" Required height= "image-height" Required uri= "*relative path to location of image*" Required imagtype= image/jpg| image/gif| image/png Required x0=x-offset when positioning a background image in a drag and drop y0=y-offset
matemtext	material	Used to define a link which opens a popup. Is a child of <material> either in a question or in one of the responses to a question or in the feedback to a question.	uri="relative path to location of html page" width="width of pop-up window" height="height of pop-up window" Text contained in this tag becomes the text of the link.
mattext	material	Used to define the text part of <material> as part of a question, or response option or feedback.	Label=|question| caption Optional, used only to specify if <mattext> is part of a question or an image caption in cases where it could be ambiguous.
Metadata	Unit-of-study or Knowledge-	Used only to include <Title> for units-of-study and	

	object	knowledge objects.	
Method	Unit-of-study	Container for <activity-structure> in module struct.xml files in which the navigation between units of study in the module is defined.	
Objective-description	Learning-objective	contains description of a single learning objective, inside <P> element.	
Objective-type	Learning-objective	One of the restricted set of objectives, included as an empty element. (attitude, competence, insight, knowledge, other, skill)	
Other	Objective-type or Prerequisite-type	Used as an empty element.	
P	Objective-description or Prerequisite-description or Source	Container for paragraphs of text.	
Play	Method	Container for the <Activity-structure-ref>	
Pre-requisite	Prerequisites	Container for a single <Pre-requisite>	
Prerequisite-description	Pre-requisite	Contains a description of a single pre-requisite within a <P> element.	
Prerequisites	Unit-of-study	Container for a set of pre-requisites.	
Prerequisite-type	Pre-requisite	One of the restricted set of objectives, included as an empty element. (attitude, competence, insight, knowledge, other, skill, situational-factor)	
presentation	item	Used as a container in which a question can be presented. There should be one <presentation> in each <item>. The presentation area can be defined.	width=width of area height = height f area x0=x-offset of area y0=y-offset of area Optional
questestinterop	Interactions	Used as a container for questions in the form of <item>	
render_choice	response_lid	Used to contain the responses in a choice type question (e.g multiple choice or matrix)	
render_extension	response_grp	Used to contain the response labels in a drag and drop type of question.	
render_hotspot	response_lid	Used to contain the responses in a hotspot type question.	
response_grp	presentation	Used as a container to group draggables or targets	ident=”draggables|targets”

		together into groups in a drag and drop type of question.	
response_label	render_choice	Used to define each response to the question that the student can select. In "choice" questions this will usually be <material>. In a hotspot it will be an rarea defined as a polygon (bounded: list of x,y coordinate pairs), circle (ellipse: x, y of centre and radius) or rectangle (x and y .coordinates of two opposite corners). In the case of drag and drop questions it will be a match_group when linking draggables to drop areas but is also used to define the coordinates of drop zones (x,y,width,height). Only rectangular drop zones are possible. If only x,y is provided then target is a point and a target icon appears (usually in text drag and drops).	ident="*unique label which is also used by <itemfeedback> to link them together*" labelrefid=correct|incorrect Required rarea=Bounded| Ellipse| Rectangle Required, but only for hotspot questions. match_group="*comma-separated list of labels of target <response_labels>*" Required but only for drag and drop questions
response_lid	item	Used to contain the responses to the question. The rcardinality attribute defines if only one or multiple answers are acceptable.	ident=??? Not used? Optional rcardinality=Single|Multiple Required
Roles	Unit-of-study	Used for consistency with EML but all module struct.xml files should contain only <Learner Id="learner" Max-persons="1" Min-persons="1"/>	
Role-ref	Play	Included only for consistency with EML.	Id-ref = "learner" Required and only option supported.
Section	Source	Used to break the <Source> into sections.	
Send-mail	Communication-object	Used only to offer a "hook" which could be used in the future.	
Situational-factor	Prerequisite-type	Used as an empty element.	
Skill	Objective-type or Prerequisite-type	Used as an empty element.	
Source	Knowledge-object or Section	Container for all the main presentation of information in a knowledge object.	
Special	Source	Used to include an object which cannot be included by other means (e.g. a java	Content-type="java applet|Flash" Link-name="*relative path*

		applet or Flash animation)	*to the file containing the code which will be included verbatim*" Required Position = "new-flow" Optional
Synchronous-conference	Communication-object	Used only to offer a "hook" which could be used in the future.	
Title	Metadata	Used to provide a title for knowledge objects and units-of-study – not required	
Unit-of-study	Root element, has no parent	Root element in struct.xml files.	Link-name = "*text to be used as title for unit-of-study*"
Unit-of-study-ref	Activity-sequence or Activity-selection	Used to define the unit-of-study to be presented when selected through navigation. Only used in module struct.xml files.	Link-name = "*text to be used as a title for the unit-of study*" Ref-worldwide-unique-id = "*relative path to the unit-of-study struct.xml file*" popup: is not an attribute but a processing instruction. <?popup contains the text to appear in a popup window to explain the context of the unit-of-study in this module ?>

Appendix C: Installing MeteoCAL directory structure and tools

1. These instructions assume that a Java Runtime environment is already installed. If not then download it from http://java.sun.com/j2se/1.4.1/download.html and install it.

2. Copy all files on the CD to your hard drive. You can put them wherever you wish. It is best to choose a final location, as the configuration will need to change of you later decide to move the files. When copying files from a CD they are always copied “read-only”. To make these files editable you need to reset this property. To do this use the “Find” or “Search” facility to find all the files you have copied. This can be done by searching for all files (*.*) in the directory where you copied them. Make sure “search subfolders” is selected. This search should find nearly 5000 files. Once the files have been found, press CTRL-A to select them all then right-click the mouse over the selected files and ensure that the “read-only” property is not set. All files are now ready to be modified.

3. Note that the MSGCAL directory is all you really need. Other directories are supplied only to support this course and are not essential.

4. Throughout this course\ and its installation you may find it useful to have an editor available on the desktop which you can drag and drop files onto. For example you may want to have *notepad* (usually in the Windows directory) or *wordpad* (usually in the Program files/Accessories or Program files/Windows NT/Accessories directories) available. Create shortcuts for one or both of these and place them on the desktop.

5. To set up intramediate (the MSG-CAL processor) go to directory msgcal\tools\intramediate. In the bin directory there are files intramediate.bat (for use on PCs) and intramediate.sh (for use on Unix systems).
    a. For Windows 95/98/ME PCs only (for other opeartying system go immediately to 5b): Use the intramediate_win9x.bat file. Additionally, edit the intramediate_win9x.bat file, and set the INTRAMEDIATE_HOME variable to the path of the intramediate directory, e.g. “set INTRAMEDIATE_HOME=D:\msgcal\intramediate”
    b. Right-click on intramediate.bat (Windows NT/2000/XP) or intramediate_win9x.bat (Windows 95/98/ME) and "create a shortcut". You may like to rename the shortcut to something like "MSGCAL processor" but this is not necessary. Drag the shortcut to place it on your desktop.

6. Next, you need to provide information to intramediate about where the MSGCAL directories are on your machine (Note that if you move these directories, for example to a server, you will need to repeat this stage.) Open the file intramediate.xml which you will find in the intramediate\resources directory - you can open this file to edit it in any editor you normally use, for example notepad, WordPad or word. (If you have placed one of these editors on the desktop then you need only drag and drop the intramediate.xml file on the editor.) This file contains:

```
<intramediate>
  <projectRoot>///barbar/apache/htdocs/devel/msgcal/</projectRoot>
  <debug>false</debug>
</intramediate>
```

7. The first and last lines simply identify this as the container for intramediate settings and should not be modified. The second line should be changed to show the full path of the MSGCAL directory on your machine. For someone unfamiliar with the operation of intramediate it is also advisable to change "false" on the third line to "true" in order to ensure that debugging information is switched on. Make sure you save the file with its original name and “.xml” extension. A modified file might look like:

```
<intramediate>
<projectRoot>C:\msgcal\</projectRoot>
```

```
<debug>true</debug>
</intramediate>
```

8. Installation and configuration of the tools is now complete. To test that the processor is operating correctly find the file msgcal\source\English\msg\calib\conclusions\quiz.xml and drag it and drop it over the “msgcal processor” on the desktop. If everything is working you will see a window appear with the following:

```
command line
args:D:\cd_march_2003\msgcal\source\english\msg\calib\conclusions\quiz.xml
passing [D:\cd_march_2003\msgcal\source\english\msg\calib\conclusions\quiz.xml] as
argument.
about to create: D:\cd_march_2003\msgcal\www\english\msg\calib\conclusions\quiz.htm
have just created: D:\cd_march_2003\msgcal\www\english\msg\calib\conclusions\quiz.htm-
--true
Completed.
Press any key to continue . . .
```

Press any key and that window will disappear. If error messages appear check appendix D for sources of common errors. This process has created a new html file which is located at msgcal\www\english\msg\calib\conclusions\quiz.htm and you can check that this has happened by checking the date and time that file was updated.

Appendix D: Common errors when processing XML files and their causes

***Error**: Out of environment space*

Cause: This occurs on some Windows 9x systems. A full explanation and remedy is described at http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q230205&. The short answer is to edit the c:\config.sys file on your machine (using sysedit if available), and add the following line:

SHELL=C:\COMMAND.COM /E:4096 /P

***Error**: incomplete arguments were given*

Cause: The version of intramediate you are using needs to have the "transformation type" specified. To produce a www version of the MSGCAL, add the parameter www to the line in the intramediate.bat file as shown below:

java -classpath %CLASSPATH% com.intrallect.intramediate.Intramediate www %CMD_LINE_ARGS%

***Error**: Specified project root does not exist*

Cause: The MSGCAL directory cannot be found. Check if you have the correct path in the intramediate\resources\intramediate.xml file.

***Warning**: The element type "element-name" must be terminated by a matching end tag*

Cause : The XML you are processing is not well formed, in that the tags are not all opened and closed in the correct order. If you have a validating XML editor such as XML Spy or Xeena, it should help you find the problem.

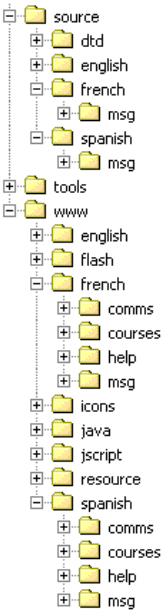
Appendix E: Localisation – Adding a new language

MeteoCAL was originally designed to have two languages, English and French, and to switch between them. However it is also possible to add a completely new language, at least for languages which use a Latin character set. This appendix describes all the files which need to be copied, created, edited or translated to introduce a new language. The next appendix describes how to make that new language available to users.

There are two aspects to adding a new language: creating modules in a new language; making the interface operate in a new language. These are dealt with separately.

Creating modules in a new language

To create modules in a new language you need to create new directories for the new language in both the **source** and **www** directories. In the example shown the **french** directory has been copied and pasted and renamed **spanish** in both the **source** and **www** directories. Creating modules in the new language is simple. You only need to create the source material in appropriate files in the source directory and process them in the usual way. The HTML files will be processed into the correct language directory in **www**.



The source form of files is always held in units of study within modules:

msgcal/source/*language*/msg/*module*/*unit-of-study*/

and the equivalent HTML files are in

msgcal/www/*language*/msg/*module*/*unit-of-study*/

Remember that resources are language independent so they all go in the **www/resource** directory irrespective of language of the module.

When you are creating modules you will also add new material to the reference library. Since the contents of the reference library for any unit-of-study are defined in the struct.xml source file for that unit-of-study you can make the reference library for one language different from the reference library for another – you may have different library material in different languages. If you want to include reference library material in a mixture of languages (for example include English material in the Spanish reference library) then you must place those English files in the www/spanish/msg/library/ directory, as well as in the www/english/msg/library/.

Best practice: Use UTF-8 for the encoding of your XML (MeteoCAL) source files (although ISO-8859-1 can be also processed)

Making the interface operate in a new language

The interface has several different elements to it. It includes the following components and the relevant files all need to be translated into the new language:

Interface component	**Files to be modified/translated**
Course information	www/*language*[10]/courses/msgcrs/index.htm www/*language*/courses/msgcrs/crsindex.htm
Buttons, tool tips, etc	tools/xsl/language/language.xml www/*language*/language.js
Glossary	www/*language*/msg/glossary/pindex.htm
Communications modes	www/*language*/comms/pindex.htm
Help information	www/*language*/help/pindex.htm

Now each of these files will be considered in detail to identify the changes that need to be made.

Changing www/*language*/courses/msgcrs/index.htm

This is described in detail in the following appendix.

[10] Where *language* appears in italics you should replace the word with the new language you are adding, e.g. spanish.

Changing www/*language*/courses/msgcrs/crsindex.htm
This file describes the course and offers an introduction to the available modules. Use an existing language file as a template and translate the content. Remember to translate “Module Index” and any module titles as well.

Changing tools/xsl/language/language.xml
It is important to realise that this is not a language specific file – it contains all the languages supported by the MeteoCAL tools. When modifying this file you should add to it but do not remove, or change, other languages. This file contains a number of sections such as:

```
<langitem name="_ANIMATION_GO_TO_START">
<langstring language="english">Go to start of animation</langstring>
<langstring language="french">Aller au début de l'animation</langstring>
</langitem>
```

In each and every section you should add the equivalent term in the new language, such as:

```
<langitem name="_ANIMATION_GO_TO_START">
<langstring language="english">Go to start of animation</langstring>
<langstring language="french">Aller au début de l'animation</langstring>
<langstring language="spanish">Ir al principio de la animación</langstring>
</langitem>
```

Changing www/*language*/language.js
Unlike the previous example, this is a language specific file. To modify this file you copy a file from an existing language, such as English or French and translate the appropriate terms. For example, based on the French file which contains:

```
var prevPageText = "Page précédente"
var nextPageText = "Page suivante"
var modStrucText = "Structure du module"
var refLibText = "Bibliothèque de référence"
```

You could create a Spanish file containing:

```
var prevPageText = "Página anterior"
var nextPageText = "Página siguiente"
var modStrucText = "Estructura del módulo"
var refLibText = "Biblioteca de consulta"
```

Note that all terms in the file must be translated.

Changing www/*language*/msg/glossary/pindex.htm, www/*language*/comms/pindex.htm and www/*language*/help/pindex.htm
These are ordinary HTML file and you should translate the content but leave all the HTML markup unchanged. Note that images (for example in the help file) will be picked up automatically from the language independent resource directory.

Appendix F: Localisation – Setting which languages will be displayed

Meteo-CAL provides a bilingual interface as standard. New languages can be added – see the previous appendix for details. A powerful feature allows users to switch from one language to the other for each individual page of the course.

The instructions below describe how to modify the switching between languages.

Changing www/*language*/courses/msgcrs/index.htm
Near the top of this file is a section that looks like:

```
<SCRIPT LANGUAGE="javascript">
<!--
//set languages here
var language1 = "english";
var language2 = "french";
var contentPage="crsindex.htm"
//-->
</script>
```

You need to modify the settings for language1, the default language, and language2, the alternative language. So a course in Spanish with English as the alternative language should appear as:

```
<SCRIPT LANGUAGE="javascript">
<!--
//set languages here
var language1 = "spanish";
var language2 = "english";
var contentPage="crsindex.htm"
//-->
</script>
```

The functionality of the language swap can be extended to include 3 languages (and, in fact, you could extrapolate from these instructions to accommodate even more languages). For example, you might like your students to be able to move from English to Spanish to French (and then back to English). To achieve this do the following: In index.htm in the course directory (in our case, msgcrs) in the english directory structure, set language1 = “english” and language2 = “spanish”. In the equivalent file in the spanish directory set language1 = “spanish” and language2 = “french”. To complete the loop, in the equivalent file in the spanish directory structure set language1 = “french” and language2 = “english”.

If you do not wish your course to feature more than one language (perhaps you do not have all the modules in more than one language or it is just not appropriate) then you can do this. If you set language2=”null” in the same location discussed above, the change language button will not appear in the interface.

Notice that the language names should not have capital letters as they should match the names of the language directories.

Appendix G: General Internationalisation/Localisation Guidelines

The aim of this document is to list all parameters to consider during the MSG training library development, to ensure that the modules and the client infrastructure will suit the needs of people of different background and different cultural origin, and that all contents can be easily translated later, without any technical skills.

This document can be used as a reminder by initial authors. Its final version will be used also later during the maintenance of the existing modules or creation of new ones.

In this document, general considerations are first presented for each item, *and concrete solutions in the frame of the MSG training library development are listed in italics.*
In the following texts, the author is a generic name for the team writing the story-board, drafting the contents and developing the module. The evaluator is a generic name for all people participating at any stage at the selection of story-boards, the evaluation of the pedagogical contents, ease of use of the modules. The user is the generic name for the students, the meteorologists and researcher who will study the modules.
The tutor is the generic name for the teachers, presenters, on-line tutors etc.. who will use the library to teach or present people the corresponding information.

Contents

This first –non technical- issue is not the simplest. It is extremely difficult for an author to identify how he/she is biased by his/her own culture, the scientific culture of his/her institution (University, NMS, organisation). The solution is the implementation of a continuous cycle of development / evaluation, to allow early identification of items which not suit all possible users.

At all stages of the development cycle, the help of the evaluators is sought, for story-board selection or content reading. It is important to involve future users of the Anglo-Saxon, Latin and African countrNon textual-contents
All meteorological images, graphics, illustrations etc… are mainly originating from one NMS or University, hence often presented in a special environment.

As far as possible, specific elements (for instance forecaster workstation look and feel) are removed, or, if not possible, explained for a foreign user.

Pedagogy

"Training adults in work-related activities is best achieved through “situated learning”. This places the training directly in the context of the work-place by using training activities which are as close to the everyday work activities as possible. In the case of the MSG CAL modules this will be achieved by using problem-solving tasks directly appropriate to operational weather forecast offices." (MSG Proposal)

From Scandinavia to Africa, the forecasters can perform very different tasks.

A questionnaire can be sent to a sample of future users to ask them to check what are their daily activities. A fast track in the modules can then include common activities, while more specific tasks could be addressed in second level pages.

Concerning the pedagogic approach, modularity, interactivity, different tracks within the modules help users to find an adapted pedagogical path, or tutors to adapt them to fit their student's culture.

Interface

A close attention should be given to all elements, colours, icons, to avoid cultural bias.

A sample of the interface is sent to a selection of people of different origins to ask them if the icons are clear.

To help user to learn in different languages, or tutors to swap for a multilingual audience, a language choice is available at any moment to swap to another language at the very same position.
A generic icon for language swap will open a menu offering available languages

The help files are provided in the language of the application.
Special attention is given to the glossaries, which can be accessed in the alphabetical order of each language.

If support material (glossary, reference library) is not in the module's language, a link will be nevertheless provided indicating the language(s) of this material.

Icon do not have lingual elements, but an ALT tag and roll-over description in the status bar will be associated with each icon in each language

Architecture

To allow future easy translation, it is important to separate lingual contents from non lingual content. All relative resource paths should be identical for each language, to avoid any link correction after translation of the material and installation in a new directory. The corresponding technical solution is presented in the section "architecture' of the general report.

A language directory will contain only the following formats: XML and its applications (SVG, etc…)

A resource directory will contain the following format : XML, GIF, JPG, SWF, SVG and a few others like movie files etc.

All lingual contents of the complete library are included in XML files, so that they can be parsed for immediate translation.

XML coding

The XML files will be coded in UTF-8, as the currently most supported coding of Unicode, used for example in the Java .class files, and the default coding used by a XML processor.

Though this is not necessary, the XML declaration will include encoding="UTF-8".

UTF-8 including ASCII, the 7-bit ASCII files can be correct XML files, without any declaration.

XML tags

XML tagging in a training application can be classified into 2 classes: the tags relevant to pedagogy and the tags relevant to the subject matter.

The naming of the XML element types themselves is biased towards one language – likely English - which is unavoidable, but a careful attention is given to the choice of these names, and a central document keeps all element types and the associated concepts in, at least, English and French, and ready to localise in other languages.

We will then support two documents:

- The DTD, which will be "in English" or better, whose element names, restricted vocabulary etc…) are in English.
- The Description of the element names, in English and (in a first step) in French

Example of the Description document, French version

<ELEMENT satellite (Meteosat | MTP | MSG)>

L'élément <satellite> qui décrit le type de satellite peut avoir les trois valeurs suivantes :
"Meteosat" = Météosat

"Meteosat transition programme" = ¨Programme de Transition Météosat
"MSG" = Météosat Seconde Génération

A meteorological and scientific classification, or vocabulary, will be developed, at least when a concrete pedagogical goal is foreseen (allowing for instance to tag what is satellite technical information, pure meteorological content, product oriented, image oriented etc…). This allows to produce modules for different audiences. This classification will be internationalised (by checking that all concepts are multi-cultural) and localised in two languages.

Graphics

Current standards (GIF and JPEG) are allowed for graphics without lingual elements.

For graphics with lingual elements, the use of a vectorial, XML based solution is necessary. The decision shall be taken very early.

SVG is a good solution, as it delivers XML directly to the parser, and does not need to point to source documents for possible translation. Nevertheless, current browsers do not all process SVG.

Flash can be also used, with the shortcoming that the XML source is kept separately, and not necessarily delivered to the end user.

Animation

An XML based solution will be defined for the animations. The point is that a non-lingual animation has no internationalisation issues, but if the animation is to have a caption for each image (what we call a slide show) then the file names and the captions can be stored in XML.

Reference Library

Reference material is either already available, or provided by other people, without any content modification by the authors.

This reference material will be also translated into XML and a nomenclature will be created for quick reference through the application.

As any learning object, the reference documents will be identified with LOM and indicate if other lingual versions are available. This will allow the training application to link to the reference material and indicate if it is available in the language of the module or only in other languages.

Communication tools

"The communication will offer the use of internet-based communication with experts" (MSG Proposal)

Concrete cases :

Synchronous communication can be implement for specific events, where tutors know the lingual background of the users.

The most frequent case will be the use of plain asynchronous email. The communication tools of the MSG training library will include a special facility, using the following CEN/ISSS draft CWA, allowing the users to specify their language capability profile, which is made of one or several language capability definition.
This will allow the tutor or responsible of the training to transmit the message to the relevant expert.

As no tracking of an individual user is foreseen, the user will have to fill the language profile form (a few click) each time he asks for information.

Language capability definition (data model) :

Nr.	Name	Explanation	Size	Value Space	Data type	Example
1	Language		1	Repertoire of ISO 639-1	Charstring	fr-CA
2	Communication direction	The combination of this data element with the following describes all types of skills : for instance output+text = write		input output	Charstring	
3	Communication mode			oral text other (braille...)		
4	Level	**No knowledge**: if necessary to emphasise in particular cases **Basics**: skills to understand the general meaning of a text or to perform basic communication **Medium**: skills necessary to live in the corresponding country **Good:** able to discuss, understand newspaper or tv, write correct texts **Excellent** : perfect mastery in the language		no-knowledge basics medium good excellent		

The following is an example of the XML syntax which could be used for temporary storage during a session:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE profile [
<!ELEMENT profile (definition+)>
<!ELEMENT definition (direction, mode, level)>
<!ATTLIST definition language CDATA #required>
<!ELEMENT direction (empty)>
<!ATTLIST direction type (input | output) #required>
<!ELEMENT mode (empty)>
<!ATTLIST mode type (oral | text | other) #required>
<!ELEMENT level (empty)>
<!ATTLIST level type (no-knowledge | basics | medium| good |
excellen) #required>
]>
<profile>
    <definition language="it-IT">
      <direction type="input"/>
      <mode type="text"/>
      <level type="good"/>
   </definition>
    <definition language="es-AR">
      <direction type="input"/>
      <mode type="oral"/>
      <level type="medium"/>
   </definition>
</profile>
```

Metadata

The selected metadata scheme shall reflect the multilingual aspects of a training library.

LOM is a good solution, with the elements "IsVersionOf" or "HasVersion" allowing to track the initial lingual version of a module and the possible alternative lingual versions.

Java

Java already has a full internationalisation system. All lingual dependencies are stored in the parameters file which is read at run-time when the language is specified. All languages are stored in a single file making it easy to translate (but hard for use to store in separate lingual directory structures.